目　　录

第1部分　基础理论

第2部分　时频分析工具箱及其应用

第1部分　基础理论

在许多工程应用场合，信号往往是非平稳的，其统计量是时变函数，这时仅仅了解信号在时域或频域的全局特性是远远不够的，需要知道信号频谱随时间变化的情况。为了分析和处理非平稳信号，人们对傅立叶分析进行了推广甚至根本性的革命，提出并发展了一系列新的信号处理理论，联合时频分析就是其中一种重要的方法。联合时频分析的基本思想是设计时间和频率的联合函数，利用它可以同时描述信号在不同时间和频率的能量密度和强度。目前时频分析方法已经广泛应用于通信、自动化、雷达、声纳、生物、天文、医学、地球物理和故障诊断等几乎所有技术领域。

常用的信号时频分析方法包括短时傅立叶变换、Wigner-Ville 分布和小波变换，本书第一部分将对这些基本理论知识进行简单介绍，更详细的论述建议读者查阅相关文献和书籍。

本部分内容主要包括：

- **时频分析的基本理论；**
- **Matlab 简介。**

第 1 章　时频分析基本理论

利用时间和频率的联合函数来表示信号简称为信号的时频表示，时频表示分为线性和二次型两种。典型的线性时频表示有：短时傅立叶变换和小波变换。在很多实际场合，还需要二次型的时频表示来描述该信号的能量密度分布，称之为信号的时频分布，典型的是 Wigner-Ville 分布。

本章主要内容包括：

- 短时傅立叶变换；
- 小波变换；
- Wigner-Ville 分布；
- Matlab 简介。

1.1　时频分析的基本概念

1.1.1　非平稳随机信号

随机信号在理论上可以分为平稳和非平稳两大类。长期以来，由于理论研究和分析工具的局限，人们常常将非平稳信号简化为平稳信号进行处理。但是严格说来，许多实际中的信号都是非平稳信号。

所谓非平稳信号是指统计特性随时间变化的随机信号。一般来说，非平稳随机信号的概率密度 $p(t,x)$ 是时间的函数。在 $t=t_i$ 处，其概率密度函数仍满足

$$\int_{-\infty}^{\infty} p(x,t_i)dx = 1 \tag{1.1.1}$$

以 $p(t,x)$ 为基础可以定义均值 $m_x(t)$、均方值 $D_x(t)$ 和方差 $\sigma_x^2(t)$ 如下：

$$\begin{aligned} m_x(t) &= E[x(t)] = \int_{-\infty}^{\infty} xp(x,t)dx \\ D_x(t) &= E[x^2(t)] = \int_{-\infty}^{\infty} x^2 p(x,t)dx \\ \sigma_x^2(t) &= D_x(t) - m_x^2(t) \end{aligned} \tag{1.1.2}$$

式中 E[・]表示期望。需要特别指出的是，对非平稳信号而言，其统计特性只能在集平均上有意义，没有时间平均意义上的统计特征。

常用的非平稳信号的自相关函数和功率谱定义如下：

$$\begin{aligned} R_x(t,\tau) &= E[x(t+\frac{\tau}{2})x^*(t-\frac{\tau}{2})] \\ S_x(t,f) &= \int_{-\infty}^{\infty} R_x(t,\tau)e^{-j2\pi f\tau}d\tau \end{aligned} \tag{1.1.3}$$

式中 τ 表示延迟，f 表示频率。

1.1.2　时频分析的由来及分类

时频分析方法是非平稳信号处理的一个重要分支，它是利用时间和频率的联合函数来表示非平稳信号，并对其进行分析和处理。

经典的分析与处理平稳信号的最常用、最主要的方法是傅立叶变换，它建立了信号从时域到频域的变换桥梁，而傅立叶反变换则建立了信号从频域到时域的变换桥梁，它们之间是一一对应的映射关系，可以用下式来表示

$$X(f)=\int_{-\infty}^{\infty}x(t)e^{-j2\pi ft}dt \tag{1.1.4a}$$

$$x(t)=\int_{-\infty}^{\infty}X(f)e^{j2\pi ft}df \tag{1.1.4b}$$

因此在传统的信号分析预处理中，时域和频域构成了表征信号的两种主要方式。但是傅立叶变换的不足在于它是在整体上将信号分解为不同的频率分量，从而缺乏局域性信息，即对信号的表征要么完全在时域，要么完全在频域，它不能揭示某种频率分量出现在什么时候以及随时间的变化情况。

为了克服传统傅立叶变换的这种全局性变换的局限性，必须使用局部变换的方法，用时间和频率的联合函数来表示信号，这就是时频分析思想的来源。时频分析方法按照时频联合函数的不同可以分为线性时频表示和双线性时频表示两种。

（1）线性时频表示

线性时频表示是由傅立叶变换演化而来的，满足线性叠加性。假设 $x(t)=ax_1(t)+bx_2(t)$，记 $x(t), x_1(t), x_2(t)$ 的线性时频表示分别为 $P(t,f), P_1(t,f), P_2(t,f)$，则有

$$P(t,f)=aP_1(t,f)+bP_2(t,f) \tag{1.1.5}$$

常见的线性时频表示主要有短时傅立叶变换、Gabor 展开以及小波变换等。短时傅立叶变换实质上是加窗的傅立叶变换，随着时间窗的移动而形成信号的一种时频表示。Gabor 展开是最早提出的一种时频表示，它可以看作是短时傅立叶变换在时域和频域进行取样的结果。对短时傅立叶变换和 Gabor 展开来说，窗函数的宽度是固定的，而小波变换是一种窗函数宽度可调的时频表示。

（2）双线性时频表示

双线性时频表示也称作二次型时频表示，它反映的是信号能量的时频分布。二次型时频表示不满足线性叠加性。假设 $x(t)=ax_1(t)+bx_2(t)$，记 $x(t), x_1(t), x_2(t)$ 的线性时频表示分别为 $P(t,f)$, $P_1(t,f), P_2(t,f)$，则有

$$P(t,f)=|a|^2P_1(t,f)+|b|^2P_2(t,f)+2\mathrm{R}[abP_{12}(t,f)] \tag{1.1.6}$$

式中最后一项称之为干扰项，这是二次型时频表示固有的一个属性。

二次型时频表示主要有 Cohen 类时频分布和仿射类（Affine）双线性时频分布，其中最著名的是 Wigner-Ville 分布。

1.1.3　瞬时频率和群延迟

令 $s(t)$ 是一实的非平稳信号，在进行时频分析之前，往往需要先将实信号 $s(t)$ 转变为复信号 $z(t)$ 的形式，最简单的方法是利用 Hilbert 变换定义实信号 $s(t)$ 的复信号。

定义 1.1 （解析信号）假设 $x(t)$ 是一实的非平稳信号，则其解析信号 $z(t)$ 定义为

$$z(t)=s(t)+j\mathrm{H}[x(t)] \tag{1.1.7}$$

式中 $\mathrm{H}[x(t)]$ 是 $x(t)$ 的 Hilbert 变换。

解析信号的优点在于它剔除了实信号中的负频率成分，同时不会造成任何信息损失，也不会带来虚假信息。

在非平稳信号分析中，瞬时物理量往往起着重要的作用，瞬时频率和群延迟就是这样的两个物理量。

（1）瞬时频率

对一个具有有限能量的复信号 $z(t)=A(t)e^{j\phi(t)}$（$A(t)$为实函数）来说，可以定义其瞬时频率为相位函数对时间的导数，即

$$f(t)=\frac{1}{2\pi}\frac{d\phi(t)}{dt} \tag{1.1.8}$$

上式的物理意义是十分明显的，并且可以证明：信号谱的平均频率等于瞬时频率的时间平均。

（2）群延迟

与时域信号 $z(t)$ 的瞬时频率相对应，频域信号的 $Z(f)$ 的群延迟 $g(f)$ 也是一个重要的瞬时参数，它表示频谱 $Z(f)$ 中频率为 f 的各个分量的延迟，定义为

$$g(f)=-\frac{1}{2\pi}\frac{d\arg Z(f)}{df} \tag{1.1.9}$$

离散信号 $z(n)$ 的群延迟定义为

$$g(k)=\frac{1}{4\pi}[\arg Z(k+1)-\arg Z(k-1)] \tag{1.1.10}$$

和瞬时频率一样，群延迟也有自己的物理解释。

1.1.4 信号分辨率

在非平稳信号处理中，窗函数常常起着关键的作用，所加的窗函数能否正确反映信号的时频特性，也即窗函数是否具有高的时间分辨率和频率分辨率，与待分析信号的非平稳特性有关。

（1）时间分辨率

对于信号 $x(t)$ 而言，其能量的密度函数可以记作 $|x(t)|^2$，在 Δt 内的能量为 $|x(t)|^2\Delta t$，而信号总能量 E_1 可以表示为

$$\mathrm{E}_1=\int_{-\infty}^{\infty}|x(t)|^2\,dt \tag{1.1.11}$$

由上式可见，利用 $x(t)$ 可以确切知道每个时间点的能量密度，因此可以说信号的时间函数具有无限的时间分辨率，而信号频谱函数的分辨率为零。

（2）频率分辨率

对频谱为 $X(f)$ 的信号，其信号能量的密度函数可以记作 $|X(f)|^2$，即能量密度函数。在 Δf 内的能量为 $|X(f)|^2\Delta f$，而信号总能量 E_2 可以表示为

$$\mathrm{E}_2=\int_{-\infty}^{\infty}|X(f)|^2\,df \tag{1.1.12}$$

由 $X(f)$可以确切知道每个频率点的能量密度，因此可以说信号的频谱函数具有无限的频率分辨率，而信号时间函数的分辨率为零。

（3）不确定性原理

定义 1.2 （不确定性原理）对于有限能量的任意信号，其时宽和频宽的乘积总是满足下面的不等式

$$T \times B = \Delta t \times \Delta f \geq \frac{1}{4\pi} \tag{1.1.13}$$

式中Δt,Δf分别称作时间分辨率和频率分辨率。

不确定性原理的重要意义告诉我们：既有任意小的时宽，又有任意小的频宽的窗函数是不存在的。也即所有的时频表示只能不同程度地近似表示信号在(t,f)的能量密度。

1.2　短时傅立叶变换

为了处理时域和频域的局部化矛盾，人们提出了所谓的“时域局部化方法”。典型的是1946 年 Gabor 提出的短时傅立叶分析方法，其基本思想是：傅立叶分析是频域分析的基本工具，为了达到时域上的局部化，在信号傅立叶变换前乘上一个时间有限的窗函数，并假定非平稳信号在分析窗的短时间隔内是平稳的，通过窗在时间轴上的移动从而使信号逐段进入被分析状态，这样就可以得到信号的一组“局部”频谱，从不同时刻“局部”频谱的差异上，便可以得到信号的时变特性。

1.2.1　基本概念

定义 1.3 （短时傅立叶变换）给定一个时间宽度很短的窗函数$\eta(t)$，让窗滑动，则信号$z(t)$的短时傅立叶变换（STFT）定义为

$$\mathrm{STFT}_z(t,f) = \int_{-\infty}^{\infty} z(t')\eta^*(t'-t)e^{-j2\pi f t'}dt' \tag{1.2.1}$$

由式可见，正是由于窗函数$\eta(t)$的存在，使得短时傅立叶变换具有了局域特性，它既是时间的函数，也是频率的函数。对于给定的时间 t，$\mathrm{STFT}_z(t,f)$可看作是该时刻的频谱。特别地，当窗函数取$\eta(t)\equiv 1,\forall t$ 时，则短时傅立叶变换退化为传统的傅立叶变换。

定义式（1.2.1）表明，信号 $z(t')$在时间 t 处的短时傅立叶变换就是信号乘上一个以 t 为中心的“分析窗”$\eta(t'-t)$后所作的傅立叶变换。因为信号 $z(t')$乘以一个短窗函数$\eta(t'-t)$等价于取出信号在分析时间点 t 附近的一个切片，所以短时傅立叶变换 $\mathrm{STFT}_z(t,f)$可以理解为信号 $z(t')$在时间点 t 附近的傅立叶变换，即“局部频谱”。

窗函数$\eta(t)$的中心 $E\{\eta(t)\}$和半径$\Delta(\eta)$分别定义为

$$E\{\eta(t)\} = \frac{\int_{-\infty}^{\infty} t\,|\eta(t)|^2\,dt}{\|\eta\|_2^2} \tag{1.2.2}$$

$$\Delta\{\eta\} = \sqrt{\frac{\int_{-\infty}^{\infty}(t-E\{\eta\}^2)^2\,|\eta(t)|^2\,dt}{\|\eta\|_2^2}} \tag{1.2.3}$$

窗函数$\eta(t)$的宽度为 $2\Delta(\eta)$。

根据定义，短时傅立叶变换得到时窗$[E\{\eta\}+t-\Delta\{\eta\},E\{\eta\}+t+\Delta\{\eta\}]$和频窗$[E\{\Gamma\}+f-\Delta\{\Gamma\},E\{\Gamma\}+f-\Delta\{\Gamma\}]$中信号的局部信息。选定窗函数$\eta(t)$之后，这个时频窗是一个与两坐标轴平行、与时间 t 和频率 f 无关的矩形，具有固定的面积 $4\Delta\{\eta\}\Delta\{\Gamma\}$。短时傅立叶变换的时频分析能力可以用时频窗矩形的形状和面积来度量：在时频窗的形状固定不变时，窗口面积越小，说明时频局部化描述能力就越强；窗口面积越大，说明时频局部化描述能力就越差。

为了使短时傅立叶变换真正成为一种有实际价值的非平稳信号分析工具，信号 $z(t)$应该能够由 $\mathrm{STFT}_z(t,f)$完全重构出来。

定义 1.4 （短时傅立叶变换的完全重构条件）记重构窗函数为 $g(t)$，则要求窗函数满足以下条件:

$$\int_{-\infty}^{\infty}\eta^*(t)g(t)dt=1 \tag{1.2.4}$$

证明如下：假设重构公式为

$$\beta(u)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\mathrm{STFT}_z(t,f)g(u-t)e^{j2\pi fu}dtdf \tag{1.2.5}$$

将式（1.2.1）代入式（1.2.5），可以得到

$$\begin{aligned}\beta(u)&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\{\int_{-\infty}^{\infty}z(t')\eta^*(t'-t)e^{-j2\pi ft'}dt'\}g(u-t)e^{j2\pi fu}dtdf\\&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\{\int_{-\infty}^{\infty}e^{-j2\pi f(t'-u)}df\}z(t')\eta^*(t'-t)g(u-t)dt'dt\\&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}z(t')\eta^*(t'-t)g(u-t)\delta(t'-u)dt'dt\\&=z(u)\int_{-\infty}^{\infty}\eta^*(u-t)g(u-t)dt\\&=z(u)\int_{-\infty}^{\infty}\eta^*(t)g(t)dt\end{aligned} \tag{1.2.6}$$

因此，为了完全重构信号，即使得$\beta(u)=z(u)$，则必须满足条件

$$\int_{-\infty}^{\infty}\eta^*(t)g(t)dt=1$$

证毕。

完全重构条件是一个很宽松的约束条件，对于一给定的分析窗函数$\eta(t)$，满足条件式（1.2.4）的窗函数 $g(t)$有无穷种可能的选择。特别地，我们感兴趣的是取 $g(t)=\eta(t)$，此时对应的完全重构条件式（1.2.4）变为

$$\int_{-\infty}^{\infty}\eta^*(t)\eta(t)dt=\int_{-\infty}^{\infty}|\eta(t)|^2\,dt=1 \tag{1.2.7}$$

即所谓能量归一化，此时式（1.2.5）可写成

$$z(u)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\mathrm{STFT}_z(t,f)\eta(u-t)e^{j2\pi fu}dtdf \tag{1.2.8}$$

式（1.2.8）称之为广义短时傅立叶变换。与正、反傅立叶变换不同的是，短时傅立叶变换是一维变换，而广义短时傅立叶反变换是二维变换。

在实际应用中，需要对 $\mathrm{STFT}_z(t,f)$进行离散化处理，为此在时频面上等间隔时频网格点$(m\cdot\Delta t,n\cdot\Delta f)$处采样，其中$\Delta t,\Delta f$分别表示时间变量和频率变量的采样间隔，令 $z(k)$表示信号 $z(t)$的离散形式，则短时傅立叶变换的离散形式为

$$\mathrm{STFT}_z(m,n)=\sum_{k=-\infty}^{\infty}z(k)\eta(k\Delta t-m\Delta t)e^{-j2\pi(n\Delta f)k} \tag{1.2.9}$$

同样，式（1.2.8）可以离散为

$$z(k)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\mathrm{STFT}_z(m,n)\eta(k\Delta t-m\Delta t)e^{j2\pi(n\Delta f)k} \tag{1.2.10}$$

1.2.2　基本性质

定义式（1.2.1）表明，短时傅立叶变换是一种线性时频表示，下面分析它的基本性质。

性质 1（频移不变性）：

$$\bar{z}(t)=z(t)e^{j2\pi f_0 t}\rightarrow \mathrm{STFT}_{\bar{z}}(t,f)=\mathrm{STFT}_z(t,f-f_0) \tag{1.2.11}$$

证明如下：

$$\mathrm{STFT}_{\bar{z}}(t,f)=\int_{-\infty}^{\infty}\bar{z}(t')\eta^*(t'-t)e^{-j2\pi ft'}dt'=\int_{-\infty}^{\infty}z(t')e^{j2\pi f_0 t'}\eta^*(t'-t)e^{-j2\pi ft'}dt'$$

$$=\int_{-\infty}^{\infty}z(t')\eta^*(t'-t)e^{-j2\pi(f-f_0)t'}dt'=\mathrm{STFT}_z(t,f-f_0)$$

性质 2（不具有时移不变性）：

$$\bar{z}(t)=z(t-t_0)\rightarrow \mathrm{STFT}_{\bar{z}}(t,f)=\mathrm{STFT}_z(t-t_0,f_0)e^{-j2\pi ft_0} \tag{1.2.12}$$

证明如下：

$$\mathrm{STFT}_{\bar{z}}(t,f)=\int_{-\infty}^{\infty}\bar{z}(t')\eta^*(t'-t)e^{-j2\pi ft'}dt'=\int_{-\infty}^{\infty}z(t'-t_0)\eta^*(t'-t)e^{-j2\pi ft'}dt'$$

$$=\int_{-\infty}^{\infty}z(v)\eta^*[v+t_0-t]e^{-j2\pi f(v+t_0)}dv$$

$$=e^{-j2\pi ft_0}\int_{-\infty}^{\infty}z(v)\eta^*[v-(t-t_0)]e^{-j2\pi fv}dv$$

$$=\mathrm{STFT}_z(t-t_0,f)e^{-j2\pi ft_0}$$

由式（1.2.12）可见，通过调整相位因子，它还是可以满足时移不变性的。

性质 3（短时傅立叶变换的低通实现）：

$$\mathrm{STFT}_z(t,f)=\int_{-\infty}^{\infty}Z(f'+f)\Gamma^*(f)e^{j2\pi f't}df' \tag{1.2.13}$$

式中$\Gamma(f)$是窗函数$\eta(t)$的傅立叶变换。

证明如下：

对式（1.2.1）的右边进行传统的傅立叶变换，得到

$$F\{\mathrm{STFT}_z(t,f)\}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}z(t')\eta^*(t'-t)e^{-j2\pi ft'}dt'e^{-j2\pi f't}dt$$

$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\eta^*(t'-t)e^{-j2\pi f't}dtz(t')e^{-j2\pi ft'}dt'$$

$$=\int_{-\infty}^{\infty}e^{-j2\pi f't'}\Gamma^*(f')z(t')e^{-j2\pi ft'}dt'$$

$$=\Gamma^*(f')\int_{-\infty}^{\infty}z(t')e^{-j2\pi(f+f')t'}dt'=\Gamma^*(f')Z(f+f')$$

两边再作傅立叶反变换，得到

$$\mathrm{STFT}_z(t,f)=\int_{-\infty}^{\infty}\Gamma^*(f')Z(f+f')e^{j2\pi f't}df'$$

这就是短时傅立叶变换的低通滤波器实现，如图 1.1 所示。式（1.2.13）表明，短时傅立叶变换可以看作是$\Gamma(f')Z(f+f')$的傅立叶反变换，进一步可以解释为信号 $z(t')$经过频移f后通过频率响应为$\Gamma^*(f)$的滤波器得到的结果，这一滤波器为低通滤波器。

性质 4（短时傅立叶变换的带通实现）：

$$\mathrm{STFT}_z(t,f)=e^{-j2\pi ft}\int_{-\infty}^{\infty}Z(f')\Gamma^*(f'-f)e^{j2\pi f't}df' \tag{1.2.14}$$

证明如下：

对式（1.2.13）的积分作变量替换，则有

$$\begin{aligned}\mathrm{STFT}_z(t,f)&=\int_{-\infty}^{\infty}\Gamma^*(f')Z(f+f')e^{j2\pi f't}df'\\&=\int_{-\infty}^{\infty}Z(w)\Gamma^*(w-f)e^{j2\pi(w-f)t}dw\\&=e^{-j2\pi ft}\int_{-\infty}^{\infty}Z(f')\Gamma^*(f'-f)e^{j2\pi f't}df'\end{aligned}$$

这就是短时傅立叶变换的带通滤波器实现，如图.1.2 所示。式（1.2.14）表明，短时傅立叶变换可以理解为加窗谱 $Z(f')\Gamma(f'-f)$的傅立叶反变换，进一步可以解释为信号 $Z(f')$通过频率响应为$\Gamma^*(f'-f)$的滤波器输出乘以 $e^{-j2\pi ft}$得到的结果，这个滤波器是一个带通滤波器，其中心频率为分析频率f。

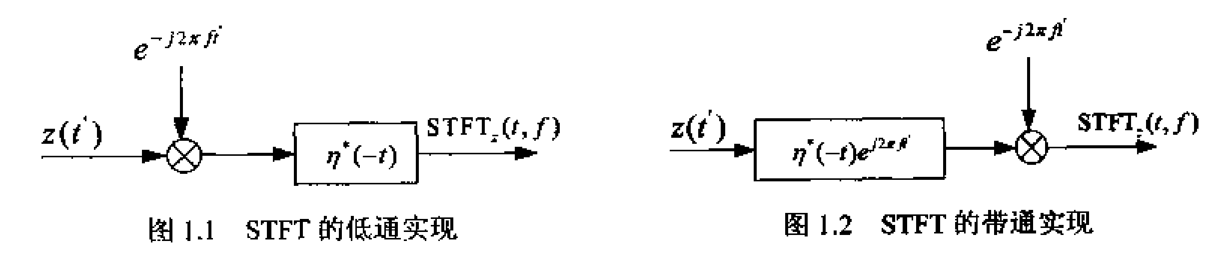


图 1.1　STFT 的低通实现　　　　图 1.2　STFT 的带通实现

1.2.3　窗函数的选择

根据定义式（1.2.1），信号 $z(t)$在时间 t 处的短时傅立叶变换是对 $z(t')$预加窗函数$\eta(t'-t)$后的频谱，故以时间 t 为中心的局部窗时间宽度内的所有信号性质都会在 $\mathrm{STFT}_z(t,f)$内显示出来，所以窗函数对短时傅立叶变换的性能有很大的影响。

显而易见，为了提高短时傅立叶变换的时间分辨率，需要选择的窗函数$\eta(t)$尽可能短。另一方面，短时傅立叶变换要得到高的频率分辨率，要求选择的窗函数$\eta(t)$时间宽度尽可能长，因此与时间分辨率的提高相矛盾。实际中，选择的窗函数$\eta(t)$的宽度应该与信号的局域平稳长度相适应。

下面说明 $g(t)=\eta(t)$时如何选择窗函数$\eta(t)$，记 $L^2(R)$表示为可测量的、模平方可积分的一维函数空间。原则上窗函数$\eta(t)$可以在 $L^2(R)$空间内任意选择，但是在实际中，我们常常要求选择的窗函数具有很好的时间和频率聚集性，使得 $\mathrm{STFT}_z(t,f)$能够有效地表征信号 $z(t)$在时频点(t,f)附近的特性，也即$\eta(t)$的窗宽度应该与信号的局部平稳长度相适应。

常用的一种窗函数选择为高斯窗函数，即

$$\eta(t)=2^{1/4}e^{-\pi t^2} \tag{1.2.15}$$

由其得到的基函数 $\eta^0_{t,f}(t')=\eta^0(t'-t)e^{j2\pi ft'}$ 就是工程中常称作的 Gabor 基函数。Gabor 基函数在时频平面上高度集中在时频点(t,f)附近，即

$$\int_{-\infty}^{\infty}u\,|\eta^0_{t,f}(u)|^2\,du=t \tag{1.2.16}$$

$$\int_{-\infty}^{\infty}f'\,|\Gamma^0_{t,f}(f')|^2\,df'=f \tag{1.2.17}$$

窗函数与局部平稳长度之间的关系表明，时频分析适合于处理局部平稳长度大的非平稳

信号。如果局部平稳长度很小，则时频分析的效果较差。

但根据 Heisenberg 不确定性原理，时间分辨率与频率分辨率不能同时任意小，它们的乘积受到一定值的限制。要提高时间分辨率就要降低频率分辨率，反之亦然。实际应用中一旦窗函数选定，则时间分辨率与频率分辨率确定，但变化着的不同时间段的信号只能加相同的窗，所以它不适应信号频率高低变化的不同要求，因此具有不变窗的短时傅立叶变换，更适合应用在准稳态信号分析的场合。另外，时间越长，信号的“局部”平稳性就越难保证，这些是短时傅立叶分析方法的不足之处。

短时傅立叶变换虽然可以描述某一局部时间段上的频率信息，但是其时、频域的分辨率$\Delta t,\Delta f$不随时间 t 和频率f的变化而变化。对于我们要分析的非平稳信号而言，也许某一小时间段上是以高频信息为主，我们希望用短时间窗进行分析；而在某一长时间段上是一些低频信息，我们希望用一个长时间窗进行分析。因此对一个时变的非平稳信号，利用短时傅立叶变换方法难以找到一个合适的时间窗口来适应于不同的时间段，这也是它的最大不足之处。

1.3 小 波 变 换

小波变换是 20 世纪 80 年代后期发展起来的一门新兴的应用数学分支，近年有学者将小波变换引入到工程振动信号分析等领域中。在理论上，构成小波变换比较系统框架的主要是法国数学家 Y.Meyer、地质物理学家 J.Morlet 和理论物理学家 A.Grossman 的贡献。而把这一理论引用到工程应用，特别是信号处理领域，法国学者 I.Daubechies 和 S.Mallat 发挥了极为重要的作用。在工程应用领域，特别在信号处理、图像处理、语音分析、模式识别和量子物理等领域，小波变换被认为是信号分析工具和方法上的重大突破。

小波变换具有多分辨特性，也叫多尺度特性，可以由粗到精地逐步观察信号，也可以看成是用一组带通滤波器对信号作滤波。通过适当地选择尺度因子和平移因子，可得到一个伸缩窗，只要适当地选择基本小波，就可以使小波变换在时域和频域都具有表征信号局部特征的能力，基于多分辨分析与滤波器组相结合，丰富了小波变换的理论基础，拓宽了它的应用范围，对小波滤波器组的设计提出了更系统的方法，降低了小波变换的计算量。

1.3.1 连续小波变换

1.3.1.1 定义

将任意 $L^2(R)$空间中的函数 $z(t)$在小波基下进行展开，称作函数 $z(t)$的连续小波变换（简称为 CWT），其表达式为

$$\mathrm{WT}_z(a,b)=<z(t),\varphi_{a,b}(t)>=\frac{1}{\sqrt{a}}\int_R z(t)\varphi(\frac{t-b}{a})dt \tag{1.3.1}$$

由连续小波变换的定义可知，小波变换同傅立叶变换一样，都是一种积分变换，我们称$\mathrm{WT}_z(a,b)$为小波变换系数。由于小波基不同于傅立叶基，因此小波变换与傅立叶变换有很大的不同，其中最重要的是，小波基具有尺度 a 和平移 b 两个参数。

根据时频分析的要求，构造的小波基函数$\varphi(t)$应该满足以下条件：

（1）本身是紧支撑的，即只有小的局部非零定义域，在窗口之外函数为零；

（2）本身是振荡的，具有波的性质，并且完全不含直流趋势成分，即

$$\Psi(0)=\int_{\infty}^{\infty}\varphi(t)dt=0$$

式中$\Psi(f)$是函数$\varphi(t)$的傅立叶变换。

（3）包含尺度参数 $a(a>0)$和平移参数 b

和短时傅立叶变换基一样，小波基也具有带通性质。以 $a=1$ 时的频带中心频率 f_1 和半功率带宽σ_1 为基准，有

$$f_a=f_1/a,\ \sigma_a=\sigma_1/a$$

a 增大，则时窗伸展，频宽收缩，带宽变窄，中心频率降低，而频率分辨率增高；a 减小则带宽增加，中心频率升高，时间分辨率增高而频率分辨率降低。这恰恰符合实际问题中高频信号的持续时间短、低频信号持续时间长的自然规律。因此，同固定时窗的短时傅立叶变换相比，小波变换在时频分析领域具有不可比拟的优点。

同样，连续小波变换也存在反变换。可以证明，如果采用的小波满足条件式（1.3.2），则其反变换存在。

$$C_\varphi=\int_R\frac{|\Psi(\omega)|^2}{|\omega|}d\omega<\infty \tag{1.3.2}$$

且连续小波变换的反变换表达式为

$$x(t)=\frac{1}{\int_0^{\infty}\frac{|\Psi(a\omega)|}{a}d\omega}\int_0^{\infty}\frac{da}{a^2}\int_{-\infty}^{\infty}\mathrm{WT}_z(a,b)\frac{1}{\sqrt{a}}\varphi(\frac{t-b}{a})db \tag{1.3.3}$$

1.3.1.2 基本性质

连续小波变换是一种线性变换，它具有以下几个性质。

性质 1：叠加性

假设 $x(t)$，$y(t)\in L^2(R)$空间，λ_1，λ_2 为任意常数，记 $x(t)$的连续小波变换为 $\mathrm{WT}_x(a,b)$，$y(t)$的连续小波变换为 $\mathrm{WT}_y(a,b)$，则 $z(t)=\lambda_1x(t)+\lambda_2y(t)$的连续小波变换为：

$$\mathrm{WT}_z(a,b)=\lambda_1\mathrm{WT}_x(a,b)+\lambda_2\mathrm{WT}_y(a,b) \tag{1.3.4}$$

性质 2：时移不变性

记 $x(t)$的连续小波变换为 $\mathrm{WT}_x(a,b)$，则 $x'(t)=x(t-t_0)$的连续小波变换为

$$\mathrm{WT}_{x'}(a,b)=\mathrm{WT}_x(a,b-t_0) \tag{1.3.5}$$

此性质说明，延时后的信号的小波系数可将原信号的小波系数在 b 轴上进行相应时移即可。

性质 3：尺度转换

记 $x(t)$的连续小波变换为 $\mathrm{WT}_x(a,b)$，则 $x'(t)=x(\frac{t}{\rho})$（$\rho>0$ 为常数）的连续小波变换为

$$\mathrm{WT}_{x'}(a,b)=\sqrt{\rho}\mathrm{WT}_x(\frac{a}{\rho},\frac{b}{\rho}) \tag{1.3.6}$$

此性质说明，当信号在时域作某一倍数伸缩时，其小波变换在 a,b 两轴上也作同一倍数伸缩，且形状不变。

性质 4：内积定理

假设 $x(t),y(t)\in L2(\mathrm{R})$空间，记 $x(t)$的连续小波变换为 $WTx(a,b)$，$y(t)$的连续小波变换为，则有

$$<WT_x(a,b),\ WT_y(a,b)>=C_\varphi<x(t),y(t)> \tag{1.3.7}$$

式中 C_φ是$\varphi(t)$的 Cohen 类时频分布，<>表示内积运算。

1.3.1.3　几种常用的连续小波基函数

（1）Morlet 小波

Morlet 小波是一种单频复正弦调制高斯波，其时、频域表达式分别如下：

时域：$\varphi(t)=e^{-t^2/2}e^{j\omega_0 t},\omega_0\geq 5$

频域：$\Psi(\omega)=\sqrt{2\pi}e^{-\frac{(\omega-\omega_0)^2}{2}}$

其时域和频域波形分别如图 1.3、图 1.4 所示（取ω_0=8），图中实线代表实部，虚线代表虚部。

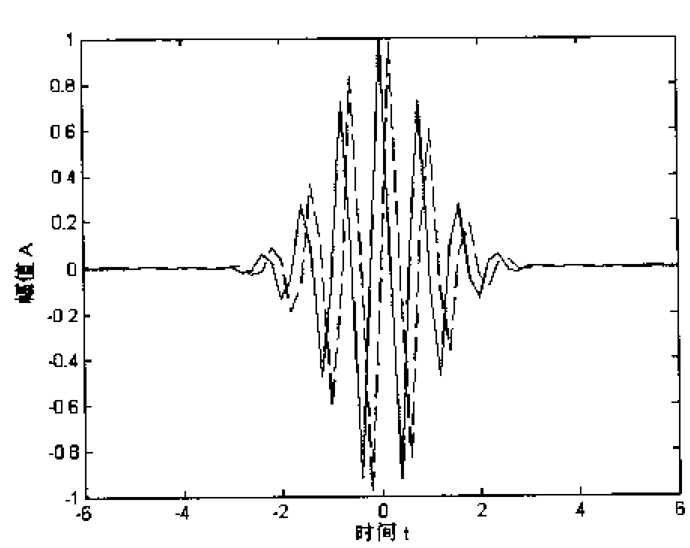


图 1.3　Morlet 小波时域波形

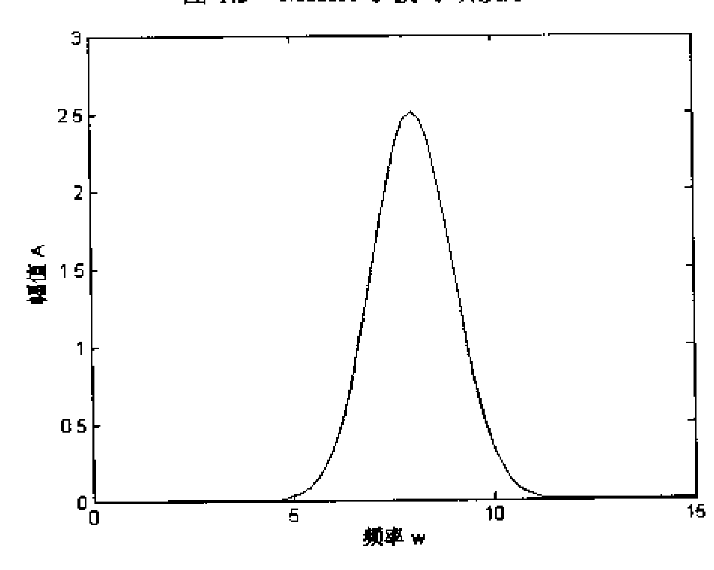


图 1.4　Morlet 小波频域波形

Morlet 小波是一种复数小波，其在时、频域都具有很好的局部性，常用于复数信号的分解及时频分析中。

（2）Marr 小波

Marr 小波的形状像墨西哥草帽，因此有时也称作墨西哥草帽小波，其时、频域表达式分别如下：

时域：$\varphi(t)=(1-t^2)e^{-t^2/2}$

频域：$\Psi(\omega)=\sqrt{2\pi}\omega^2 e^{-\frac{\omega^2}{2}}$

其时域和频域波形分别如图 1.5、图 1.6 所示。Marr 小波实际上是高斯函数的二阶导数，在时、频域都具有很好的局部性。

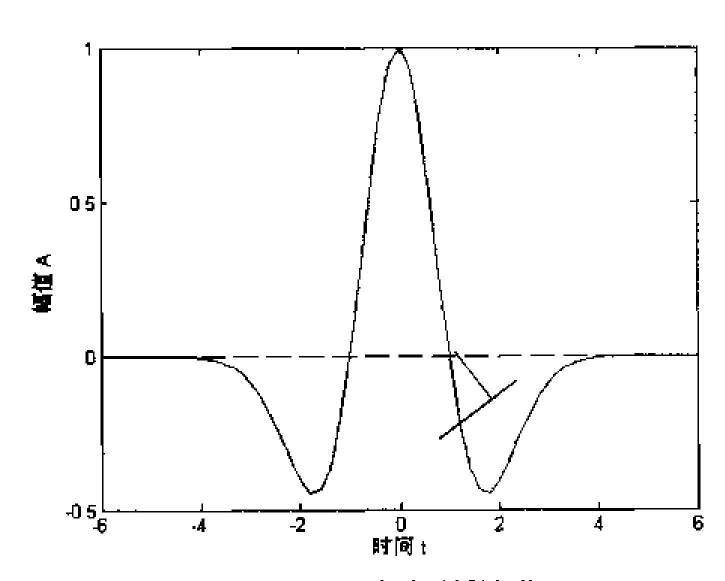


图 1.5　Marr 小波时域波形

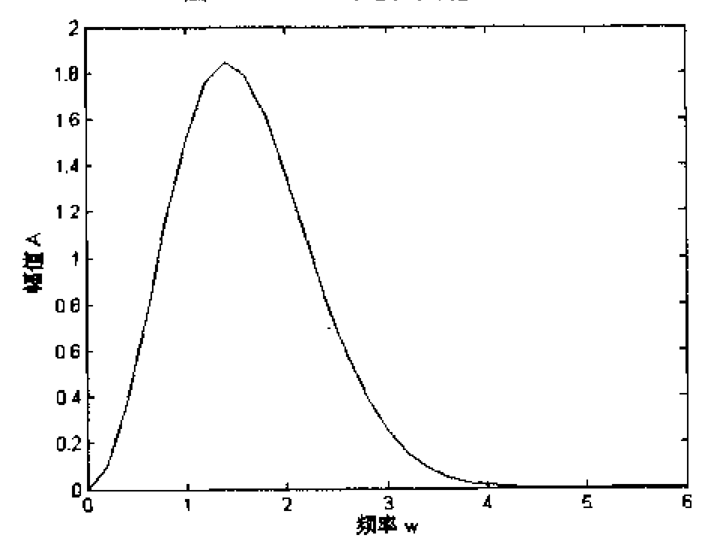


图 1.6　Marr 小波频域波形

1.3.2　离散小波变换

根据连续小波变换的定义可知，在连续变化的尺度 a 和时间 b 值下，小波基函数具有很大的相关性，因此信号的连续小波变换系数的信息量是冗余的。减小小波变换系数冗余

度的方法是将小波基函数$\varphi_{a,b}(t)$的 a,b 限定在一些离散点上取值，常用的离散化方法是将尺度按幂级数进行离散化，即取 $a_m = a_0^m$（m 为整数，$a_0 \neq 1$，一般取 $a_0=2$），对 b 进行均匀离散取值，以覆盖整个时间轴（为了不丢失信息，要求满足采样定理），这样小波基函数变为

$$\varphi_{m,n}(t)=2^{-m/2}\varphi(2^{-m}t-n)$$

因此任意函数 $z(t)$的离散小波变换为

$$\mathrm{WT}_z(m,n) = \langle z(t), \varphi_{m,n}(t) \rangle = 2^{-m/2}\int_R z(t)\varphi(2^{-m}t-n)dt$$

在离散小波中，还有一类特殊情况，即仅在尺度上进行了二进制离散，而位移仍连续变化，称之为二进小波，其表达式为

$$\varphi_{2^k,b}(t) = 2^{-k/2}\varphi(\frac{t-b}{2^k})$$

值得注意的是，离散小波变换不再具有伸缩和时移共变性质。

二进小波介于连续小波和离散小波之间，仍具有连续小波变换的时移共变性质。正因为此，它在奇异性检测、图像处理方面十分有用。

1.3.3 正交小波变换

从小波函数的条件来看，小波基不一定是正交基，但是在实际应用中希望找到正交小波基。构造正交小波基的重要方法称之为多尺度分析。

多尺度分析是通过函数空间术语来严格定义的，假设 $L^2(R)$空间内的子空间序列 $V_m(m\in Z)$ 满足以下条件：

（1）（嵌套性）$\cdots V_2 \subset V_1 \subset \cdots \subset V_{-2} \subset \cdots$；

（2）（逼近性）$\bigcap_{m\in Z} V_m = \{0\}, \bigcup_{m\in Z} V_m = L^2(R)$；

（3）（二进制伸缩性）$f(t)\in V_m \Leftrightarrow f(2t)\in V_{m-1}$；

（4）$V_m=span\{\phi_{m,n}(t), n\in Z\}$，即任一级子空间可由相应尺度的同一函数通过平移张成；

（5）$V_{m-1}=V_m \oplus W_m$，即任一级子空间可由下一级子空间以及它的正交补空间相加而成，序列 W_m 相互之间无重叠，是正交系。

这样，我们称子空间序列 V_m,W_m 为函数空间 $L^2(R)$上的一个多尺度分析，其中$\phi_{m,n}$称为尺度函数，m,n 分别是尺度和平移参数。

由以上定义可得

$$V_0 = V_m \oplus \sum_{i=1}^{m} W_i \tag{1.3.8}$$

特别地，当尺度趋于无穷时，上式变为

$$L^2(R) = \sum_{i=-\infty}^{\infty} W_i \tag{1.3.9}$$

存在与$\phi_{m,n}$相应的函数$\Theta_{m,n}$通过平移能够生成W_m。

$$W_m=span\{\Theta_{m,n}(t)\}$$

$\Theta_{m,n}$就是所求的小波基，它在伸缩和平移变换下都是正交的。

多尺度分析在信号分析中的应用可以用式（1.3.8）来表达，对于任意函数$f(t)\in V_0$，可以

在下一级尺度空间 V_1 和小波空间 W_1 上进行分解，如下所示。

$$f(t)=p_1 f(t)+q_1 f(t) \tag{1.3.10}$$

式中 $p_1 f(t)=\sum_k C_{1k}\phi_{1,k}$， $q_1 f(t)=\sum_k D_{1k}\Theta_{1,k}$ 。

$p_1 f(t)$是逼近部分，$q_1 f(t)$是细节部分。然后将逼近部分进一步分解，如此重复就可得到任意尺度上的逼近部分和细节部分，迭代公式为

$$p_{m-1}f(t)=p_m f(t)+q_m f(t)=\sum_k C_{m,k}\phi_{m,k}+\sum_k D_{m,k}\Theta_{m,k} \tag{1.3.11}$$

式中 $C_m=HC_{m-1}, D_m=GC_{m-1}$，$H$ 是低通滤波器。每一次分解，$p_m f(t)$的采样都比原来稀疏两倍，分辨率越来越粗，波形越来越光滑；G 是 H 的镜像高通滤波器，带宽每次也以两倍缩减。

经过 m 次分解，得到

$$f(t)=p_m f(t)+\sum_{j=1}^{m} q_j f(t) \tag{1.3.12}$$

式中 $p_m f(t)$给出的是函数$f(t)$的低频全局信息，第二项是逐次分解中分离得到的从 V_0 到 V_{m-1} 各个尺度上$f(t)$的相应局部细节信息。上述信号分解过程称为 Mallat 算法，其过程如图 1.7 所示。

常用的正交小波基有 Harr 小波、Littlewood-Paley 小波和 Meyer 小波。

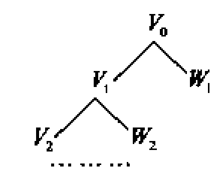


图 1.7 多尺度小波分析

1.3.4 小波包

在多尺度分析中，每一步分解都只是对尺度子空间 V_i进行，而对子空间 W_i则不再分解，因此离频跨度较宽。实际中，在很多问题中，我们只是对某些特定时间和频域段（点）的信号感兴趣，只要提取这些位置的信息即可。为此，我们希望在感兴趣的频率点上最大可能地提高频率分辨率，在感兴趣的时间点上最大程度地提高时间分辨率，此时正交小波不再满足这种要求。这需要对子空间 W_i 也一样进行分解，以提高频率分辨率，其过程如图 1.8 所示，称为小波包分解。

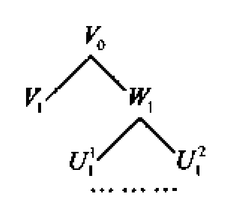


图 1.8 多尺度小波包分析

对小波子空间 W_i 可以像尺度子空间 V_i 一样利用镜像滤波器进行正交分解，对其频带进行分割。将其子空间用符号 U 统一表示，子空间分解的公式为

$$U_i^m=U_{i-1}^{2m}\oplus U_{i-1}^{2m+1} \tag{1.3.13}$$

式中下标表示尺度层次，上标区别同一层上不同的子空间。

按照式（1.3.13），小波子空间 $W_i=U_i^1$ 可以如下分解：

$$\begin{aligned}
W_i&=U_{i-1}^2\oplus U_{i-1}^3\\
W_i&=U_{i-2}^4\oplus U_{i-2}^5\oplus U_{i-2}^6\oplus U_{i-2}^7\\
W_i&=U_{i-k}^{2^k}\oplus U_{i-k}^{2^k+1}\oplus\ldots\oplus U_{i-k}^{2^{k+1}-1}
\end{aligned} \tag{1.3.14}$$

每层小波包将原频带一分为二，k 层小波包可将原频带分割为 2^k 个子频带，从而实现频带细分，提高了频域分辨率。对整个小波包来说，它是一个按二进制组织的包含从宽到窄各个频带的带通滤波器组，各种应用都能够从中找到符合需求的最优组合。

1.3.5　小波变换和傅立叶变换的比较

下面我们将以上提到小波分析方法和傅立叶变换进行比较，就更可以显示出小波变换特长所在。

（1）傅立叶变换的实质是把能量有限的信号 $f(t)$ 分解到以 $\{e^{i\omega t}\}$ 为正交基的空间上去；小波变换的实质是把能量有限信号 $f(t)$ 分解到 $W_{-j}(j=1,2,\cdots,J)$ 和 V_{-j} 所构成的空间上去。

（2）傅立叶变换用到的基本函数只有 $\sin(\omega t),\cos(\omega t),\exp(i\omega t)$，具有惟一性；小波分析所用到的小波函数则不是惟一的，同一个工程问题用不同的小波函数进行分析有时结果相差甚远。小波函数的选用是小波分析应用到实际中的一个难点问题，也是分析研究的一个热点问题，目前往往是通过经验或不断的试验，将不同的分析结果进行对照分析来选择小波函数。

（3）在频域中，傅立叶变换具有较好的局部化能力，特别是对于那些频率成分比较简单的确定性信号，傅立叶变换很容易把信号表示成各频率成分的叠加和的形式，但在时域中，傅立叶变换没有局部化能力，无法从信号 $f(t)$ 的傅立叶变换 $F(\omega)$ 中看出 $f(t)$ 在任一时间点附近的性态。

（4）在小波分析中，尺度 a 越大相当于傅立叶变换中的 ω 的值越小。

（5）在短时傅立叶变换中，变换系数 $G_f(\omega,\tau)$ 主要依赖于信号在时间窗内的情况，一旦时间窗函数确定，则分辨率也就固定了。而在小波变换中，变换系数 $WT_x(a,\tau)$ 虽然也是依赖于信号在时间窗内的情况，但时间宽度是随尺度 a 的变换而变化的，所以小波变换具有时间局部分析能力。

（6）若用信号通过滤波器来解释，小波变换与短时傅立叶变换不同之处在于：对短时傅立叶变换来说，带通滤波器的带宽 $\Delta\omega$ 与中心频率 ω 无关；相反，小波变换带通滤波器的带宽 $\Delta\omega$ 则正比于中心频率 ω，即

$$Q=\frac{\Delta\omega}{\omega}=C \qquad C\text{为常数}$$

也就是说，滤波器有一个恒定的相对带宽，称之为等 Q 结构。

1.4　Wigner–Ville 分布

从本质上说，短时傅立叶变换和小波变换都是线性时频表示，它不能描述信号的瞬时功率谱密度。此时，二次型时频表示就是一种更加直观和合理的信号表示方法，也称为时频分布，其中 Wigner-Ville 分布就是常用的一种时频分布。

1932 年，Wigner 提出了 Wigner 分布，最初应用于量子力学的研究。1948 年，Ville 将其引入信号分析领域。1970 年，Mark 提出 Wigner-Ville 分布中最主要的缺陷——交叉干扰项的存在。1980 年，Claasen 和 Mecklenbraker 联合发表的论文中详细论述了 Wigner-Ville 分布的概念、定义、性质以及数值计算等问题。

Wigner-Ville 分布是分析非平稳时变信号的重要工具，在一定程度上解决了短时傅立叶变换存在的问题。Wigner-Ville 分布的重要特点之一是具有明确的物理意义，它可被看作信号能量在时域和频域中的分布。

1.4.1 Cohen 类时频分布

20 世纪 60 年代中期，Cohen 发现众多的时频分布只是 Wigner-Ville 分布的变形，可以用统一的形式表示，习惯称之为 Cohen 类时频分布，其表达式为

$$P(t,f)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}z(u+\frac{\tau}{2})z^{*}(u-\frac{\tau}{2})\phi(\tau,v)e^{-j2\pi(tv+\tau f-uv)}dudvd\tau \tag{1.4.1}$$

式中$\phi(\tau,v)$为核函数。

由于这类分布是以核函数加权的模糊函数的二维傅立叶变换，所以 Cohen 类时频分布也称广义双线性时频分布。

1.4.2 Wigner–Ville 分布定义及性质

信号 $s(t)$的 Wigner-Ville 分布定义可以在式（1.4.1）中取核函数$\phi(\tau,v)=1$ 来定义

$$W_{z}(t,f)=\int_{-\infty}^{\infty}z(t+\frac{\tau}{2})z^{*}(t-\frac{\tau}{2})e^{-j2\pi\tau f}d\tau \tag{1.4.2}$$

式中 $z(t)$是 $s(t)$的解析信号。

Wigner-Ville 分布也可以用解析信号的频谱表示如下

$$W_{z}(t,f)=\int_{-\infty}^{\infty}Z^{*}(f+\frac{v}{2})Z(f-\frac{v}{2})e^{j2\pi tv}dv \tag{1.4.3}$$

Wigner-Ville 分布具有以下重要性质：

（1）$W_z(t,f)$对所有的 t 和 f 值是实的；

（2）$W_z(t,f)$具有时移不变性；

$$\tilde{z}(t)=z(t-t_0)\Rightarrow W_{\tilde{z}}(t,f)=W_z(t-t_0,f)$$

（3）$W_z(t,f)$具有频移不变性；

$$\tilde{z}(t)=z(t)e^{j2\pi f_0 t}\Rightarrow W_{\tilde{z}}(t,f)=W_z(t,f-f_0)$$

（4）$W_z(t,f)$满足时间边缘特性，即

$$\int_{-\infty}^{\infty}W_z(t,f)df=|z(t)|^2$$

（5）$W_z(t,f)$具有频率边缘特性

$$\int_{-\infty}^{\infty}W_z(t,f)dt=|Z(f)|^2$$

1.4.3 模糊函数

在式（1.4.1）中，记$A_z(\tau,v)=\int_{-\infty}^{\infty}z(u+\frac{\tau}{2})z^{*}(u-\frac{\tau}{2})e^{j2\pi tv}dt$，称之为模糊函数。另一方面瞬时相关函数定义为$k_z(t,\tau)=z(u+\frac{\tau}{2})z^{*}(u-\frac{\tau}{2})$，因此模糊函数可以看作是瞬时相关函数关于 t 的傅立叶反变换，即

$$A_z(\tau,v)=\underset{t\to v}{F}^{-1}[k_z(t,\tau)] \tag{1.4.4}$$

模糊函数最早应用于雷达信号分析，但采用了不同的定义，雷达模糊函数定义为：

$$A_z(\tau,v)=\underset{t\to v}{F}[k_z(t,\tau)] \tag{1.4.5}$$

对比模糊函数和 Wigner-Ville 分布可知，它们都是双线性信号的某种线性变换，前者变

换到时延-频偏平面，表示相关；而后者变换到时频平面，表示能量分布。可以证明，它们之间满足以下关系：

$$W_z(t,f)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}A_z(\tau,v)e^{-j2\pi(tv+f\tau)}dvd\tau \tag{1.4.6}$$

1.4.4　交叉项问题

虽然 Wigner-Ville 分布具有好的时频聚集性，但是对于多分量信号，根据卷积定理，其 Wigner-Ville 分布会出现交叉项，产生“虚假信号”，这也是应用中存在的主要缺陷。

交叉项是二次型时频分布的固有结果，它来自于多分量信号中不同信号分量之间的交叉作用。时频分布的交叉项一般是比较严重的，交叉项通常是振荡的，而且幅度可以达到自主项的两倍，造成信号的时频特征模糊不清。因此如何有效抑制交叉项，对时频分析非常重要。

事实上，交叉项与时频分布的有限支撑特性密切相关，而交叉项的抑制又主要通过核函数的设计来实现的。常用的加核函数后的 Wigner-Ville 分布有以下几种：

（1）伪 Wigner-Ville 分布（PWD）

$$PWD_z(t,f)=\int_{-\infty}^{\infty}z(t+\frac{\tau}{2})z^*(t-\frac{\tau}{2})h(\tau)e^{-j2\pi\tau f}d\tau=W_z(t,f)\overset{f}{*}H(f)$$

式中 $h(\tau)$是窗函数，典型的有指数函数。

（2）平滑 Wigner-Ville 分布（SWD）

$$SWD_z(t,f)=W_z(t,f)\overset{t,f}{**}G(t,f)$$

式中$\overset{t,f}{**}$表示对时间和频率的二维卷积，$G(t,f)$为平滑滤波器。

（3）平滑伪 Wigner-Ville 分布（SPWD）

$$SPWD_z(t,f)=\int_{-\infty}^{\infty}z(t-u+\frac{\tau}{2})z^*(t-u-\frac{\tau}{2})g(u)h(\tau)e^{-j2\pi\tau f}d\tau$$

式中 $g(u)$, $h(\tau)$是两个实的偶窗函数，且 $h(0)=G(0)=1$。

另外，国内外学者还研究了多种可抑制或削弱交叉项的方法，主要有：预滤波法、多分量分离法与辅助函数法，并且都采用解析信号以消除由负频率成分产生的交叉干扰项。

1.5　Matlab 简介

1.5.1　Matlab 的产生背景

Matlab 诞生在 20 世纪 70 年代，它的编写者是 Cleve Moler 博士和他的同事。当时，Cleve Moler 博士和他的同事开发了 EISPACK 和 LINPACK 的 Fortran 子程序库，这两个程序库主要是求解线性方程的程序库。但是，Cleve Moler 发现学生使用这两个程序库时有困难，主要是接口程序不好写，很费时间。于是 Cleve Moler 自己动手，在业余时间里编写了 EISPACK 和 LINPACK 的接口程序。Cleve Moler 给这个接口程序取名为 Matlab，意为矩阵（MATRIX）和实验室（LABORATORY）的组合。

1984 年，Cleve Moler 和 John Little 成立了 MathWorks 公司，正式把 Matlab 推向市场，

并继续进行 Matlab 的开发。1993 年，MathWorks 公司推出 Matlab 4.0；1995 年，MathWorks 公司推出 Matlab 4.2C 版（For Windows 3.x）；1997 年推出 Matlab 5.0；2000 年 10 月，MathWorks 公司推出 Matlab 6.0；2002 年 8 月推出 Matlab 6.5；2004 年 6 月最新的版本 Matlab 7.0 开始发布。每一次新版本的推出都使 Matlab 有了长足的进步，界面越来越友好，内容越来越丰富，功能越来越强大。

Matlab 以商品形式出现后，仅短短几年就以其良好的开放性和运行的可靠性淘汰了当时众多的软件包。进入 20 世纪 90 年代后，Matlab 已经成为国际控制界公认的标准计算软件，并在大学里广泛使用，深受大学生的喜爱。在欧美的一些大学里，诸如应用代数、数理统计、自动控制、数字信号处理、模拟与数字通信、时间序列分析、动态系统仿真等课程的教科书都把 Matlab 作为内容，在那里 Matlab 成为攻读学位的大学生、硕士生、博士生必须掌握的基本工具之一。

Matlab 长于数值计算，能处理大量的数据，而且效率比较高。MathWorks 公司在此基础上开拓了符号计算、文字处理、可视化建模和实时控制能力，增强了 Matlab 的市场竞争力，使 Matlab 成为了市场主流的数值计算软件。经过多年的工程实践，人们已经发现 Matlab 作为计算工具和科技资源，可以扩大科学研究的范围、提高工程生产的效率、缩短开发周期、加快探索步伐、激发创造活力。

1.5.2 Matlab 的强大功能

Matlab 产品组是支持概念设计、算法开发、建模仿真和实时实现的理想集成环境。无论是进行科学研究和产品开发，Matlab 产品族都是一种必不可少的工具。

Matlab 产品族可用于以下方面：

- 数据分析；
- 数值和符号计算；
- 工程与科学绘图；
- 控制系统设计；
- 数字图像信号处理；
- 财务工程；
- 建模、仿真、原型开发；
- 应用开发；
- 图形用户界面设计。

Matlab 产品族被广泛地应用于信号与图像处理、控制系统设计、通信、系统仿真等诸多领域。开放式的结构使 Matlab 产品族很容易针对特定的需求进行扩充，从而在不断深化对问题的认识同时，提高自身的竞争力。

Matlab 产品族的一大特性是有众多的面向具体应用的工具箱和仿真块，包含了完整的函数集用来对信号和图像处理、控制系统设计、神经网络等特殊应用进行分析和设计。其他的产品延伸了 Matlab 的能力，包括数据采集、报告生成和依靠 Matlab 语言编程产生独立 C/C++ 代码等。

Matlab 主要功能如下。

（1）Matlab

Matlab 是 MathWorks 公司所有产品的数值分析和图形基础环境，它将二维和三维图形、Matlab 语言能力等集成到一个单一的、易学易用的环境之中。

（2）Matlab Toolbox

工具箱是一系列专用的 Matlab 函数库，以解决特定领域的问题，它是开放的、可扩展的——用户可以查看其中的算法，或开发自己的算法。

（3）Matlab Compiler

编译器可以将 Matlab 语言编写的 m 文件自动转换成 C 或 C++文件，支持用户进行独立应用开发。结合 Mathworks 提供的 C/C++数学库和图形库，用户可以利用 Matlab 快速地开发出功能强大的独立应用。

（4）Simulink

Simulink 是结合了框图界面和交互仿真能力的非线性动态系统仿真工具，它以 Matlab 的核心数学、图形和语言为基础。

（5）Stateflow

Stateflow 与 Simulink 框图模型相结合，描述复杂事件驱动系统的逻辑行为，驱动系统在不同的模式之间进行切换。

（6）Real-Time Workshop

直接从 Simulink 框图自动生成 C 代码，用于快速原型和硬件在回路仿真，整个代码的生成可以根据需要进行完全定制。

（7）Simulink Blockset

专门为特定领域设计的 Simulink 功能块的集合，用户也可以利用已有的块或自编写的 C 和 Matlab 程序建立自己的块。

1.5.3 Matlab 语言特点

Matlab 语言有不同于其他高级语言的特点，它被称为第四代计算机语言。正如第三代计算机语言如 Fortran 语言与 C 语言等使人们摆脱了对计算机硬件的操作一样，Matlab 语言使人们从繁琐的程序代码中解放出来。它的丰富的函数使开发者无需重复编程，只要简单地调用和使用就可以了，其突出的特点是简单和直接。

Matlab 语言的主要特点有：

（1）编程效率高

Matlab 是一种面向科学与工程计算的高级语言，允许用数学形式的语言编写程序，且比 Basic、Fortran 和 C 等语言更加接近于人们书写计算公式的思维方式，用 Matlab 编写程序犹如在演算纸上排列公式和求解问题。因此，Matlab 语言也可通俗地称为演算纸式科学算法语言。由于它编写简单，所以编程效率高，易学易懂。

（2）用户使用方便

Matlab 语言是一种解释执行的语言（在没被专门的工具编译之前），它灵活、方便，其调试程序手段丰富，调试速度快，需要学习的时间少。人们用任何一种语言编写程序和调试程序一般都要经过 4 个步骤：编辑、编译、连接以及执行和调试。各个步骤之间是顺序关系，编程的过程就是在它们之间作瀑布型的循环。Matlab 语言与其他语言相比，较好地解决了上述问题，把编辑、编译、连接和执行融为一体。它能在同一画面上进行灵活操作、快速排除

输入程序中的书写错误、语法错误以及语意错误，从而加快了用户编写、修改和调试程序的速度，可以说在编程和调试过程中它是一种比 VB 还要简单的语言。

具体地说，Matlab 运行时，如直接在命令行输入 Matlab 语句（命令），包括调用 M 文件的语句，每输入一条语句，就立即对其进行处理，完成编译、连接和运行的全过程。又如，将 Matlab 源程序编辑为 M 文件，由于 Matlab 磁盘文件也是 M 文件，所以编辑后的源文件就可直接运行，而不需进行编译和连接。在运行 M 文件时，如果有错，计算机屏幕上会给出详细的出错信息，用户经修改后再执行，直到正确为止。所以可以说，Matlab 语言不仅是一种语言，广义上讲是一种语言调试系统。

（3）扩充能力强，交互性好

高版本的 Matlab 语言有丰富的库函数，在进行复杂的数学运算时可以直接调用，而且 Matlab 的库函数同用户文件在形成上一样，所以用户文件也可作为 Matlab 的库函数来调用。因而，用户可以根据自己的需要方便地建立和扩充新的库函数，以便提高 Matlab 使用效率和扩充它的功能。另外，为了充分利用 Fortran、C 等语言的资源，包括用户已编好的 Fortran，C 语言程序，通过建立 Mex 调文件的形式，混合编程，可以方便地调用有关的 Fortran，C 语言的子程序；还可以在 C 语言和 FORTRAN 语言中方便地使用 Matlab 的数值计算功能，这种良好的交互性使程序员可以使用以前编写过的程序，减少重复性工作，也使现在编写的程序具有重复利用的价值。

（4）移植性很好，开放性很好

Matlab 是用 C 语言编写的，而 C 语言的可移植性很好，所以 Matlab 可以很方便地移植到能运行 C 语言的操作平台上。适合 Matlab 的工作平台有：Windows 系列、UNIX、Linux、VMS 6.1、PowerMac。除了内部函数外，Matlab 所有的核心文件和工具箱文件都是公开的，都是可读可写的源文件，用户可以通过对源文件进行修改或自己编程构成新的工具箱。

（5）语句简单，内涵丰富

Matlab 语言中最基本、最重要的成分是函数，其一般形式为[a, b, c, …]= fun（d, e, f,…），即一个函数由函数名、输入变量 d,e,f,…和输出变量 a,b,c,…组成。同一函数名 fun，不同数目的输入变量（包括无输入变量）及不同数目的输出变量，代表着不同的含义（有点像面向对象中的多态性）。这不仅使 Matlab 的库函数功能更丰富，大大减少了需要的磁盘空间，而且使得 Matlab 编写的 M 文件简单、短小而高效。

（6）高效方便的矩阵和数组运算

Matlab 语言同 Basic、Fortran 和 C 语言一样，规定了矩阵的算术运算符、关系运算符、逻辑运算符、条件运算符及赋值运算符等，而且这些运算符大部分可以毫无改变地照搬到数组间的运算，有些如算术运算符只要增加“·”就可用于数组间的运算。另外，它不需定义数组的维数，并给出矩阵函数、特殊矩阵专门的库函数，使之在求解信号处理、建模、系统识别、控制、优化等领域的问题时，显得大为简捷、高效、方便，这是其他高级语言所不能比拟的。在此基础上，高版本的 Matlab 已逐步扩展到科学及工程计算的其他领域。

（7）方便的绘图功能

Matlab 的绘图是十分方便的，它有一系列绘图函数（命令），例如线性坐标、对数坐标，

半对数坐标及极坐标，均只需调用不同的绘图函数（命令），在图上标出图题、*xy* 轴标注，格（栅）绘制也只需调用相应的命令，简单易行。另外，在调用绘图函数时调整自变量可绘出不同颜色的点、线、复线或多重线，这种为科学研究着想的设计是通用的编程语言所不及的。



第2部分　时频分析工具箱及其应用

时频分析工具箱中提供了计算各种线性时频表示和双线性时频分布的函数，如何将这些函数用于解决实际中的工程问题是读者更关心的，为此本部分内容从具体的工程应用出发，通过实例来说明这些函数的用法。

本部分内容主要包括：

- 时频分析工具箱函数简介；
- 非平稳信号的一般处理；
- 线性时频表示；
- 二次型时频表示；
- 时频分布处理；
- 时频分析在信号处理中的应用；
- 时频分析在故障诊断中的应用。

第 2 章　时频分析工具箱函数简介

时频分析工具箱中存在大量计算各种时频表示的函数，为了掌握这些函数的基本用途，本章对这些函数进行了归纳，给出了每个函数的基本功能，具体的语法将在后面章节中使用时加以详细说明。

本章主要内容包括：

- 信号产生函数；
- 噪声产生函数；
- 线性时频处理函数；
- Cohen 类双线性时频处理函数；
- Affine 类双线性时频处理函数；
- 模糊函数；
- 其他处理函数。

表 2.1　　信号产生函数

函　数　名	功 能 说 明	函　数　名	功 能 说 明
amexpo1s	单边指数幅值调制信号	altes	时域 Altes 信号
amexpo2s	双边指数幅值调制信号	anaask	幅值键移信号
amgauss	高斯幅值调制信号	anabpsk	二进制相位键移信号
amrect	矩形幅值调制信号	anafsk	频率键移信号
amtriang	三角形幅值调制信号	anapulse	单位脉冲信号的解析投影
fmconst	定频调制信号	anaqpsk	四进制相位键移信号
fmhyp	双曲线频率调制信号	anasing	Lipschitz 奇异性
fmlin	线性频率调制信号	anastep	单位阶跃信号的解析投影
fmodany	任意频率调制信号	atoms	基本高斯元的线性组合
fmpar	抛物线频率调制信号	dopnoise	复多普勒任意信号
fmpower	幂指数频率调制信号	doppler	复多普勒信号
fmsin	正弦频率调制信号	klauder	时域 Klauder 小波
gdpower	能量律群延迟信号	mexhat	时域墨西哥帽小波

表 2.2　　噪声产生函数

函　数　名	功 能 说 明	函　数　名	功 能 说 明
noisecg	解析复高斯噪声	noisecu	解析复单位高斯噪声

表 2.3 线性时频处理函数

函 数 名	功 能 说 明	函 数 名	功 能 说 明
tfrgabor	Gabor 表示	instfreq	瞬时频率估计
tfrstft	短时傅立叶变换	sgrpdlay	群延迟估计
ifestar2	使用 AR(2)模型的瞬时频率估计		

表 2.4 Cohen 类双线性时频处理函数

函 数 名	功 能 说 明	函 数 名	功 能 说 明
tfrbj	Born-Jordan 分布	tfrri	Rihaczek 分布
tfrbud	Butterworth 分布	tfrridb	降低交叉项的分布（Bessel 窗）
tfrcw	Choi-Williams 分布	tfrridbn	降低交叉项的分布（二项式窗）
tfrgrd	归一化的矩形分布	tfrridh	降低交叉项的分布（汉宁窗）
tfrmh	Margenau-Hill 分布	tfrridt	降低交叉项的分布（三角形窗）
tfrmhs	Margenau-Hill 谱图分布	tfrsp	谱图分布
tfrmmce	谱图的最小平均互熵组合	tfrspwv	平滑伪 Wigner-Ville 分布
tfrpage	Page 分布	tfrwv	Wigner-Ville 分布
tfrpwv	伪 Wigner-Ville 分布	tfrzam	Zhao-Atlas-Marks 分布

表 2.5 Affine 类双线性时频处理函数

函 数 名	功 能 说 明	函 数 名	功 能 说 明
tfrbert	单式 Bertrand 分布	tfrspaw	平滑伪 Affine 类 Wigner 分布
tfrdfla	D-Flandrin 分布	tfrunter	Unterberger 分布
tfrscalo	尺度图		

表 2.6 模糊函数

函 数 名	功 能 说 明	函 数 名	功 能 说 明
ambifunb	窄带模糊函数	ambifuwb	宽带模糊函数

表 2.7 其他处理函数

函 数 名	功 能 说 明	函 数 名	功 能 说 明
friedman	瞬时频率密度	ridges	波峰提取
htl	图像直线检测中的 Hough 变换	plotifl	绘制归一化的瞬时频率规律
margtfr	时频表示的能量	tfrparam	返回用于显示时频表示的参数
momftfr	时频表示的频率矩	tfrqview	时频表示的快速可视化
momttfr	时频表示的时间矩	tfrsave	保存时频表示的参数
renyi	Renyi 信息度量	tfrview	时频表示的可视化

第 3 章　非平稳信号的一般处理

在非平稳信号分析中，瞬时物理量往往起着重要的作用，瞬时频率和群延迟就是这样的两个物理量。本章将介绍这两个物理量参数的计算方法，另外还将举例说明通过数值仿真生成各种非平稳信号的方法。

本章主要内容包括：

- 计算瞬时频率和群延迟；
- 非平稳信号的生成。

3.1　计算瞬时频率和群延迟

在时域和频域同时描述非平稳信号的一个重要参数是瞬时频率，它定义为解析信号相位的倒数，其物理意义表示向量幅角的转速（单位时间内转动多少周）。为了计算一个信号的瞬时频率，首先必须将分析的信号 $x(t)$转化为其解析信号 $s(t)$，常用的方法是 Hilbert 变换，即

$$s(t)=x(t)+jH[x(t)]$$

利用解析信号，可以唯一定义瞬时幅值和瞬时频率为

$$A(t)=|s(t)|$$

$$f(t)=\frac{1}{2\pi}\ \frac{d\arg s(t)}{dt}$$

时频分析工具箱中计算瞬时频率的函数为 instfreq.m。

（1）瞬时频率估计

instfreq 函数

【语法格式】

① [f,t]= instfreq(x)

② [f,t]= instfreq(x,t)

③ [f,t]= instfreq(x,t,*l*)

④ [f,t]= instfreq(x,t,*l*,trace)

【使用说明】instfreq 计算解析信号在时刻 t 处的瞬时频率。

输入参数：l 应为整数，如果 $l=1$，则按照定义 $\arg[x(t+1)x^{*}(t-1)]$来计算 x 的瞬时频率；如果 $l>1$，则计算 x 瞬时频率的极大似然估计；如果 $trace\neq0$，则跟踪算法的过程。

输出参数：f 返回归一化的瞬时频率（位于 0.0 和 0.5 之间）；t 返回时间坐标。

例程 3-1：线性调频信号的瞬时频率估计。

f3_1.m

```
%线性调频函数
sig=fmlin(256);
t=3:256;
%估计瞬时频率
ifr=instfreq(sig);
%显示结果
plot(t,ifr','LineWidth',2);
xlabel('时间 t');
ylabel('瞬时频率 f');
```

计算的结果如图 3.1 所示。

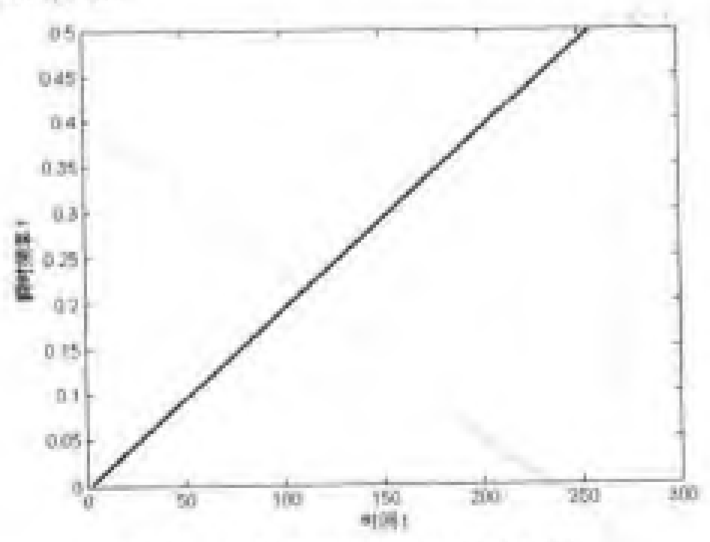


图 3.1 线性调频信号的瞬时频率估计

从图 3.1 可以看出信号频率随时间的变化曲线是一个线性关系。

瞬时频率刻画了信号随时间变化的局部频率行为，与此相对应，描述非平稳信号的另一个特征参数是群延迟，它刻画了信号随频率变化的局部时间行为，定义为

$$g(f)=-\frac{1}{2\pi}\frac{d\arg Z(f)}{df}$$

时频分析工具箱中计算群延迟的函数为 sgrpdlay.m。

（2）群延迟估计

函数名：sgrpdlay

【语法格式】

① [gd,fnorm]= sgrpdlay(x)

② [gd,fnorm]= sgrpdlay(x,form)

【使用说明】 sgrpdlay 计算信号 x 的群延迟。

输入参数：x 是分析的时域信号；form 是归一化的频率。

输出参数：gd 返回估计出的群延迟输出向量；fnorm 返回频率坐标。

例程 3-2：计算线性调频信号的群延迟

f3_2.m

```
%线性调频函数
sig=fmlin(128);
f=0:0.05:0.5;
%估计群延迟
gd=sgrpdlay(sig,f, 'LineWidth',2);
%显示结果
plot(gd,f);
xlabel('时间 t');
ylabel('归一化的频率 f');
```

计算的结果如图 3.2 所示。

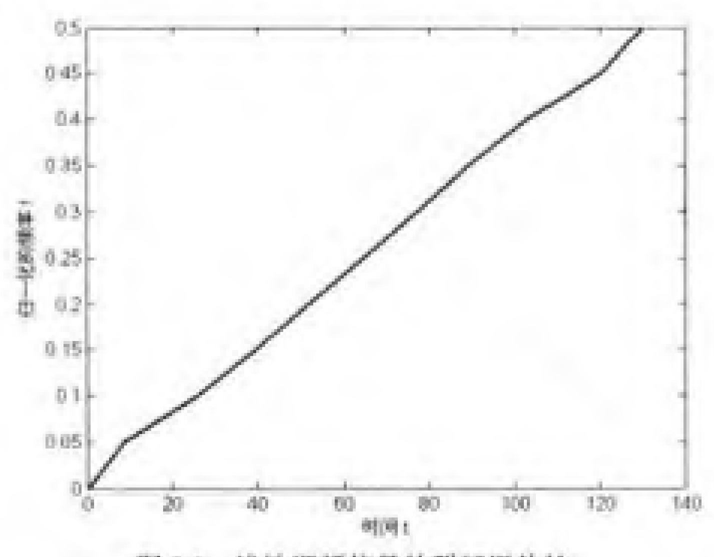


图 3.2 线性调频信号的群延迟估计

接下来，比较同一个调频信号的瞬时频率和群延迟估计。

例程 3-3：瞬时频率和群延迟估计的比较。

f3_3.m

```
%线性调频信号
sig=fmlin(256);
%频率
f=0:0.05:0.5;
%时间
t=3:256;
%瞬时频率估计
ifr=instfreq(sig);
%群延迟估计
gd=sgrpdlay(sig,f);
```

```
%显示结果
plot(t,ifr','LineWidth',2);
hold on
plot(gd,f,'o--','LineWidth',2);
legend('瞬时频率','群延迟');
xlabel('时间 t');
ylabel('归一化的频率 f');
```

计算的结果如图 3.3 所示。

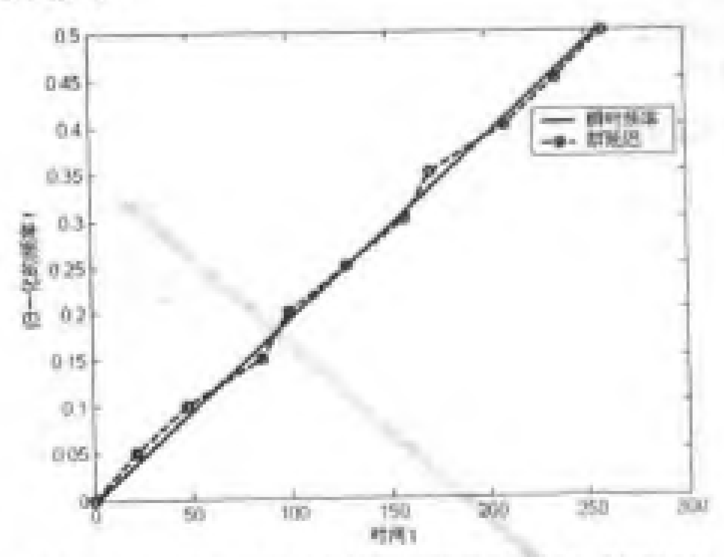


图 3.3 同一线性调频信号的瞬时频率和群延迟曲线比较

由图 3.3 可以看出，这两条曲线几乎重叠。

一般来说，瞬时频率和群延迟在时频平面内定义了两条不同的曲线，但是在时宽(T)和频宽(B)的乘积值逐渐变大时，这两条曲线会慢慢逼近。为了说明这一点，下面分别计算了两个信号的瞬时频率和群延迟，其中第一个信号的 $T\cdot B$ 值大，而第二个信号的 $T\cdot B$ 值小。

例程 3-4：$T\cdot B$ 值大的信号的瞬时频率和群延迟估计。

f3_4.m

```
%时间
t=2:255;
%信号1产生
sig1=amgauss(256,128,90).*fmlin(256,0,0.5);
%计算时宽
[tm,T1]=loctime(sig1);
%计算频宽
[fm,B1]=locfreq(sig1);
TB1=T1*B1
%计算瞬时频率
ifr1=instfreq(sig1,t);

%频率
```

```
f1=linspace(0,0.5-1/256,256);
%计算群延迟
gd1=sgrpdlay(sig1,f1);
%显示
plot(t,ifr1,'b-','LineWidth',2);
hold on;
plot(gd1,f1,'o--','LineWidth',2);
xlabel('时间 t');
ylabel('归一化的频率 f');
legend('瞬时频率','群延迟');
输出：TB1 = 13.9139。
```

计算的结果如图 3.4 所示。

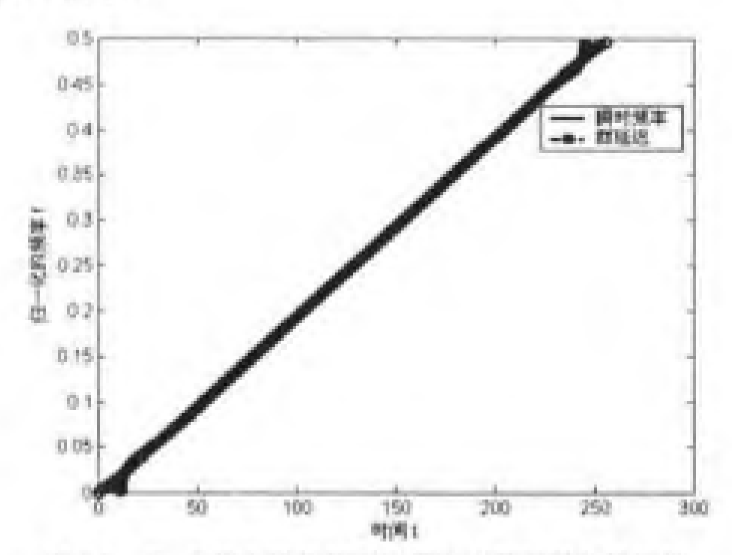


图 3.4　$T \cdot B$ 值大的信号瞬时频率和群延迟曲线比较

由图 3.4 可以看出，显然这两条曲线几乎重叠。

例程 3-5：$T \cdot B$ 值小的信号的瞬时频率和群延迟估计。

f3_5.m

```
%时间
t=2:255;
%信号 1 产生
sig2=amgauss(256,128,30).*fmlin(256,0.2,0.4);
%计算时宽
[tm,T2]=loctime(sig2);
%计算频宽
[fm,B2]=locfreq(sig2);
TB2=T2*B2
%计算瞬时频率
ifr2=instfreq(sig2,t);

%频率
```

```
f2=linspace(0.2,0.4,256);
%计算群延迟
gd2=sgrpdlay(sig2,f2);
%显示
plot(t,ifr2,'b-','LineWidth',2);
hold on;
plot(gd2,f2,'o--','LineWidth',2);
xlabel('时间 t');
ylabel('归一化的频率 f');
legend('瞬时频率','群延迟');
```

输出结果：TB2 = 1.2240。

计算的结果如图 3.5 所示。

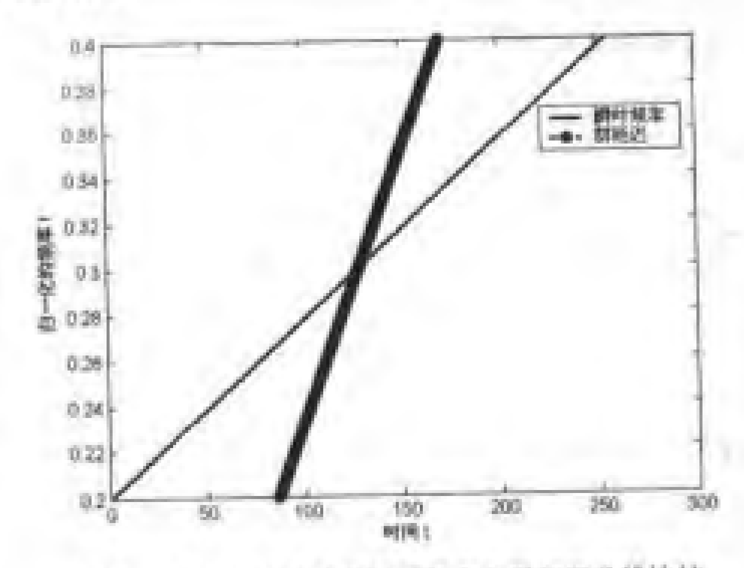


图 3.5　$T \cdot B$ 值小的信号瞬时频率和群延迟曲线比较

由图 3.5 可以看出，显然这两条曲线完全不同。

3.2　非平稳信号的生成

时频分析工具箱内提供了大量产生非平稳信号的函数，主要可以分为 3 类：

（1）第一类合成不同的幅值调制信号，这些函数都以前缀"am"开头。比如 amrect.m 产生一个矩形幅值调制信号，amgauss.m 产生一个高斯幅值调制信号等。

（2）第二类构成不同的频率调制信号，这些函数都以前缀"fm"开头。比如 fmconst.m 产生一个常数频率调制信号，fmhyp.m 产生一个双曲频率调制信号等。

（3）第三类是一系列预先定义好的信号，部分以前缀"ana"开头，因为它们自身是解析信号，而其他的都有特殊的名称。

第（1）和（2）类函数进行组合可以产生大量的非平稳信号，即将幅值调制信号乘以频率调制信号。

3.2.1 信号产生函数的基本用法

（1）第一类信号函数

1. amexpo1s 函数

【语法格式】

① y= amexpo1s (N)

② y= amexpo1s (N,t0)

③ y= amexpo1s (N,t0,T)

【使用说明】生成单边指数调幅信号。

输入参数：N 是产生的信号点数；t0 是指数函数的起始点；T 是时宽。

输出参数：y 返回产生的信号，且归一化使得 $y(t0)=1$。

例程 3-6：生成单边指数调幅信号。

f3_6.m

```
sig=amexpo1s(160);
figure(1);
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
figure(2);
sig=amexpo1s(160,20,40);
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
```

计算得到的单边指数调幅信号如图 3.6、图 3.7 所示，默认情况下指数函数的起始点位置为 t0=N/2。

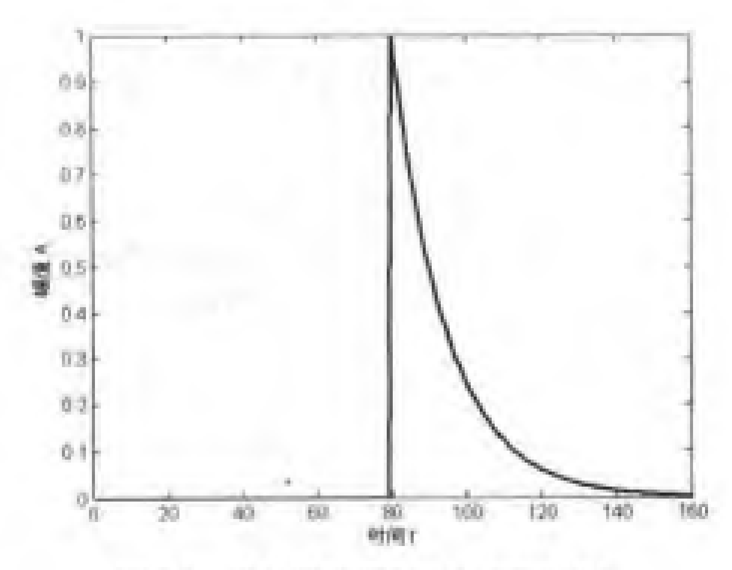


图 3.6 单边指数调幅信号（t0=N/2）

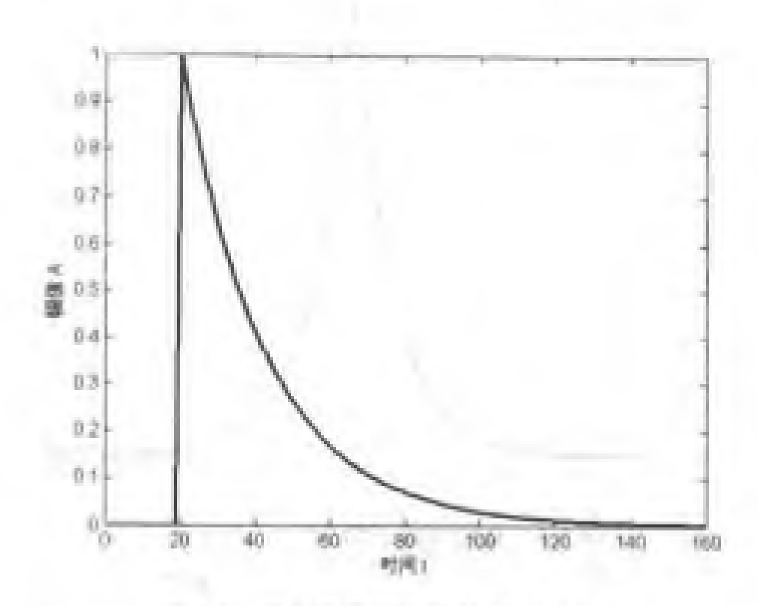


图 3.7　单边指数调幅信号（t0=20）

2. amexpo2s 函数

【语法格式】

① y= amexpo1s (N)

② y= amexpo1s (N,t0)

③ y= amexpo1s (N,t0,T)

【使用说明】生成以 t0 为中心的双边指数调幅信号。

输入参数：N 是产生的信号点数；t0 是时间中心；T 是时宽。

输出参数：y 返回产生的信号，且归一化使得 $y(t0)=1$。

例程 3-7：生成以 t0 为中心的双边指数调幅信号。

f3_7.m

```
sig=amexpo2s(160);
figure(1);
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
figure(2);
sig=amexpo2s(160,120,40);
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
```

计算得到的双边指数调幅信号如图 3.8、图 3.9 所示，默认情况下函数的中心位置为 t0=N/2。

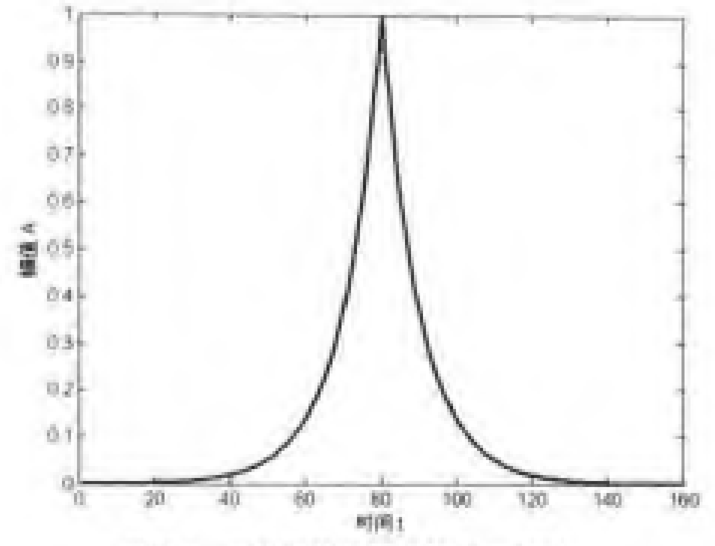


图 3.8 双边指数调幅信号（t0=N/2）

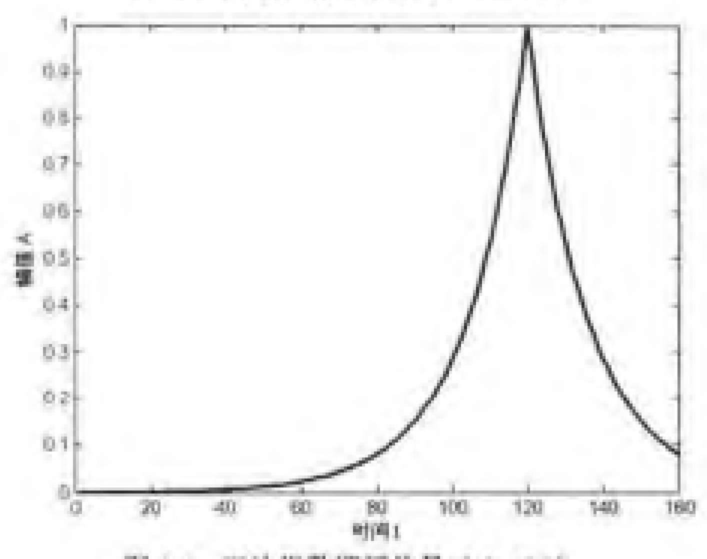


图 3.9 双边指数调幅信号（t0=120）

3. amgauss 函数

【语法格式】

① y= amgauss (N)

② y= amgauss (N,t0)

③ y= amgauss (N,t0,T)

【使用说明】生成以 t0 为中心的高斯调幅信号。

输入参数：N 是产生的信号点数；t0 是时间中心；T 是时宽。

输出参数：y 返回产生的信号，且归一化使得 y(t0)=1。

例程 3-8：生成以 t0 为中心的高斯调幅信号。

f3_8.m

```
sig=amgauss(160);
figure(1);
plot(sig,'LineWidth',2);
xlabel('时间 t');
```

```
ylabel('幅值 A');
figure(2);
sig=amgauss(160,120,40);
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
```

计算得到的高斯调幅信号如图 3.10、图 3.11 所示，默认情况下函数的中心位置为 t0=N/2。

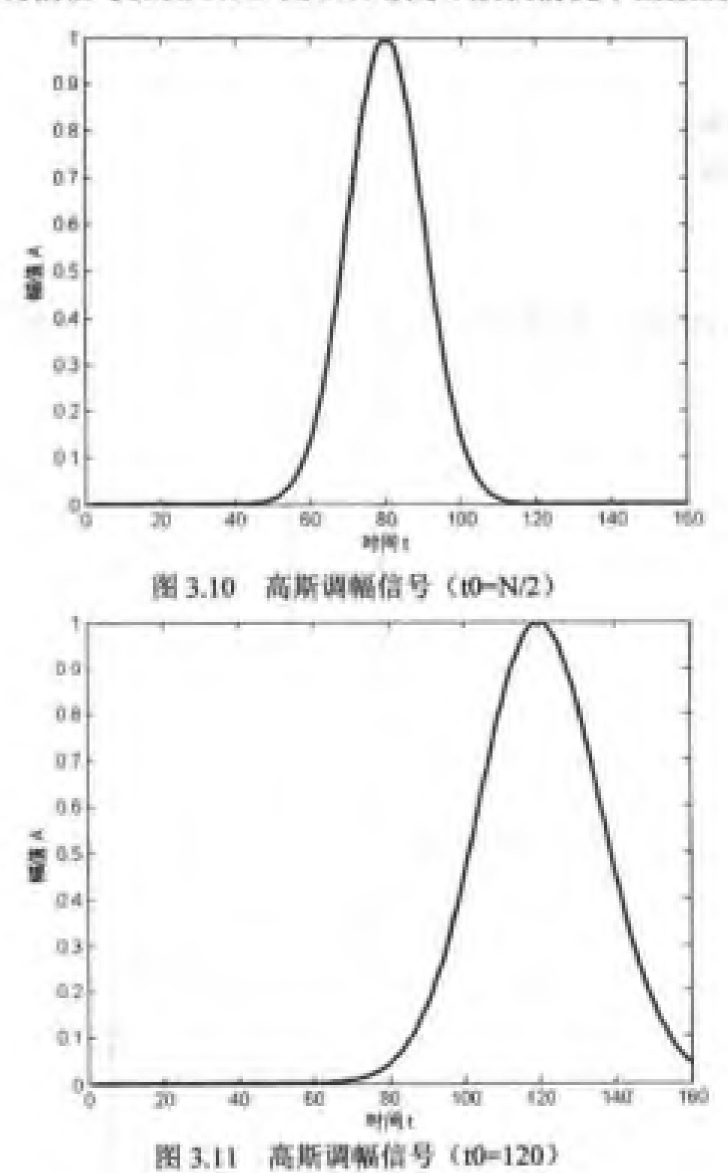


图 3.10　高斯调幅信号（t0=N/2）

图 3.11　高斯调幅信号（t0=120）

4. amrect 函数

【语法格式】

① y= amrect (N)

② y= amrect (N,t0)

③ y= amrect (N,t0,T)

【使用说明】生成以 t0 为中心的矩形调幅信号。

输入参数：N 是产生的信号点数；t0 是时间中心；T 是时宽。

输出参数：y 返回产生的信号，且归一化使得 $y(t0)=1$。

例程 3-9：生成以 t0 为中心的矩形调幅信号。

f3_9.m

```
sig=amrect(160);
figure(1);
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
figure(2);
sig=amrect(160,120,40);
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
```

计算得到的矩形调幅信号如图 3.12、图 3.13 所示，默认情况下函数的中心位置为 t0=N/2。

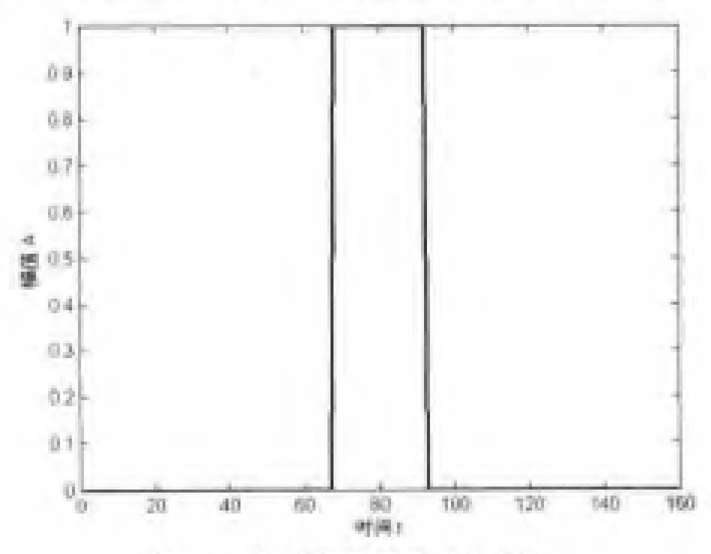


图 3.12　矩形调幅信号（t0=N/2）

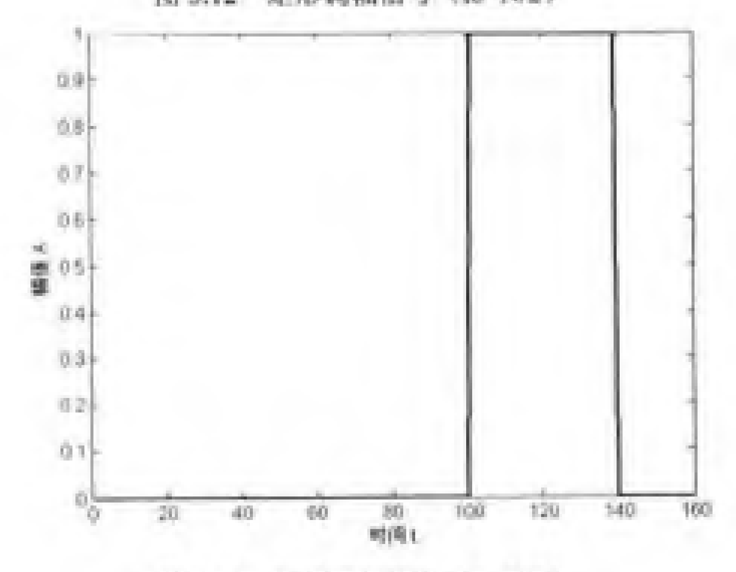


图 3.13　矩形调幅信号（t0=120）

5. amtriang 函数

【语法格式】

① y= amtriang (N)

② y= amtriang (N,t0)

③ y= amtriang (N,t0,T)

【使用说明】生成以 t0 为中心的三角形调幅信号。

输入参数：N 是产生的信号点数；t0 是时间中心；T 是时宽。

输出参数：y 返回产生的信号，且归一化使得 $y(t0)=1$。

例程 3-10：生成以 t0 为中心的三角形调幅信号。

f3_10.m

```
sig=amtriang(160);
figure(1);
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
figure(2);
sig=amtriang(160,120,40);
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
```

计算得到的三角形调幅信号如图 3.14、图 3.15 所示，默认情况下函数的中心位置为 t0=N/2。

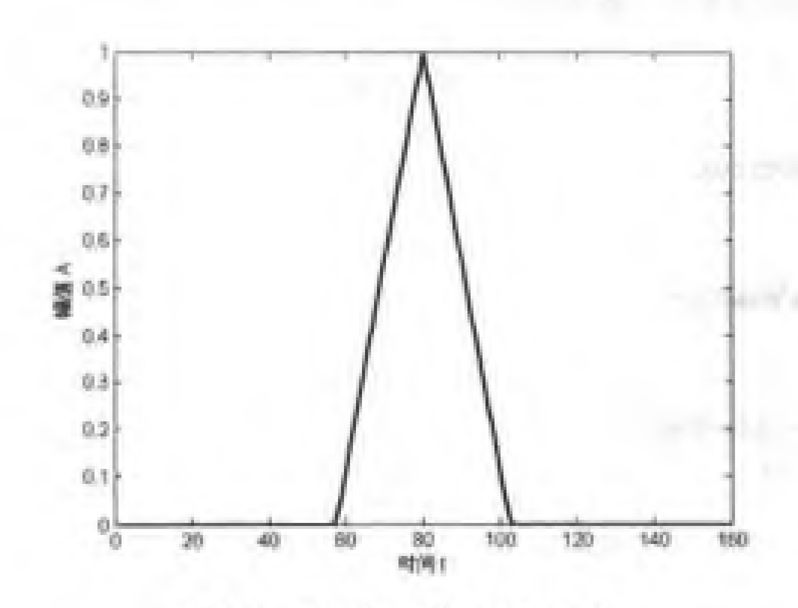


图 3.14　三角形调幅信号（t0=N/2）

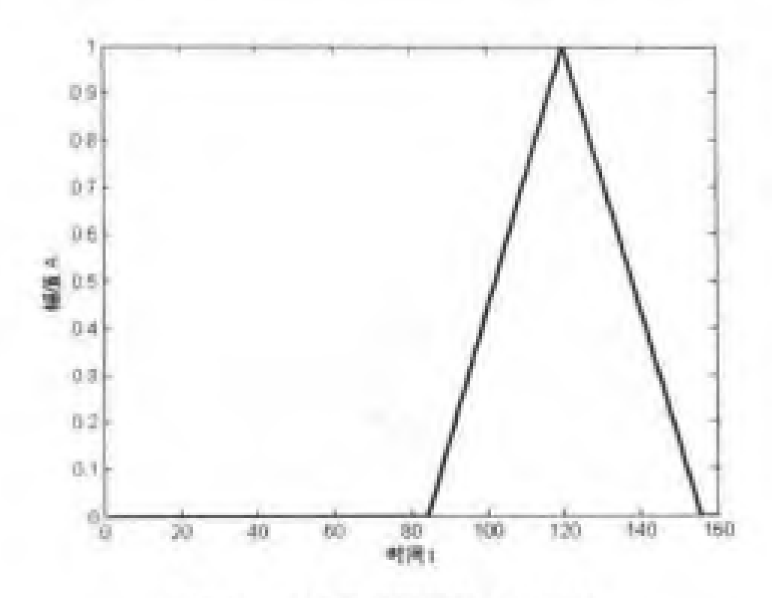


图 3.15　三角形调幅信号（t0=120）

（2）第二类信号函数

1. fmconst 函数

【语法格式】

① [y,iflaw]= fmconst(N)

② [y,iflaw]= fmconst(N,fnorm)

③ [y,iflaw]= fmconst(N,fnorm,t0)

【使用说明】生成常频率（fnorm）调制信号。

输入参数：N 是产生的信号点数；fnorm 是归一化的频率；t0 是时间中心。

输出参数：y 返回产生的信号；iflaw 为瞬时频率。

例程 3-11：生成常数频率的调制信号。

f3_11.m

```
%产生信号
sig=fmconst(128,0.05,50);
%显示
%实部
plot(real(sig),'LineWidth',2);
hold on;
%虚部
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');
```

计算的结果如图 3.16 所示，产生的信号的频率为一个常数。

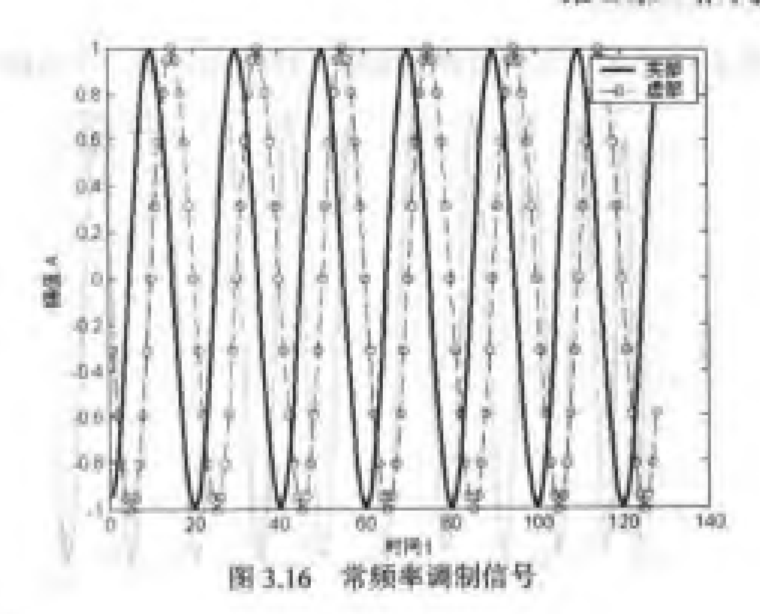


图 3.16　常频率调制信号

2. fmhyp 函数

【语法格式】

① [y,iflaw]= fmhyp (N,P1)

② [y,iflaw]= fmhyp (N,P1,P2)

【使用说明】生成双曲线频率调制信号，且

$$y(t)=\exp(j2\pi(f_0 t+\frac{c}{\lg|t|}))$$

输入参数：N 是产生的信号点数；当输入参数个数等于 2 时，P1 是含有两个系数的向量 $[f_0,c]$；当输入参数个数等于 3 时，P1(P2)是时频点$[t_i, f_i]$，且 t_i 的单位为秒，f_i 为归一化的频率；P2 同 P1。

输出参数：y 返回产生的信号；iflaw 为瞬时频率。

例程 3-12：生成双曲线频率调制信号。

f3_12.m

```
%产生信号
[sig,iflaw]=fmhyp(100,[1 0.5],[32,0.1]);
%显示
figure(1)
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');
figure(2);
plot(iflaw,'LineWidth',2);
xlabel('时间 t');
ylabel('频率 f');
```

计算的结果如图 3.17 所示，其瞬时频率如图 3.18 所示，是一个双曲线。

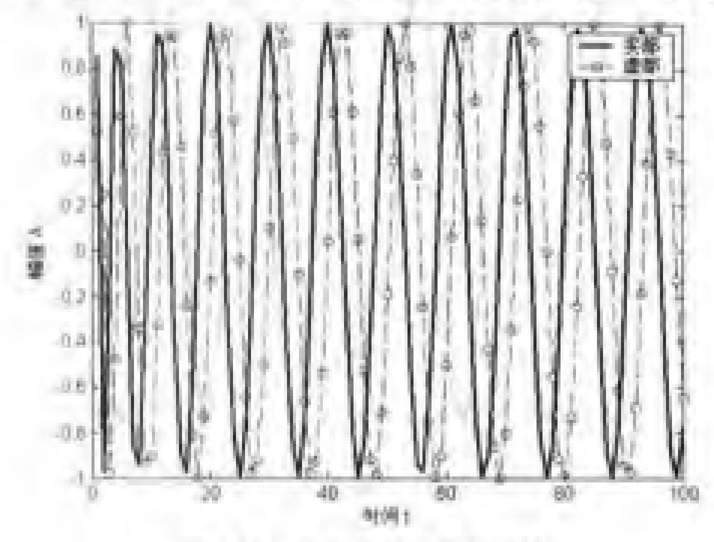


图 3.17　双曲线频率调制信号

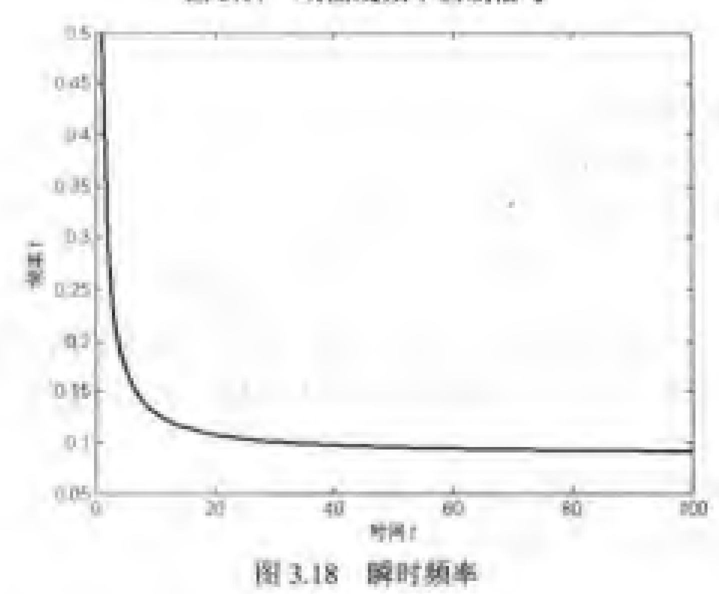


图 3.18　瞬时频率

3. fmlin 函数

【语法格式】

① [y,iflaw]= fmlin (N)

② [y,iflaw]= fmlin (N,fnormi,fnormf)

③ [y,iflaw]= fmlin (N,fnorm, fnormf ,t0)

【使用说明】生成线性调频信号。

输入参数：N 是产生的信号点数；fnormi 是初始的归一化频率；fnormf 是最后的归一化频率；t0 是相位时间参考点。

输出参数：y 返回产生的信号；iflaw 为瞬时频率。

例程 3-13：生成线性调频信号。

f3_13.m

```
%产生信号
```

```
sig=fmlin(128,0.05,0.3,50);
%显示
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');
```

计算的结果如图 3.19 所示，其频率是随时间线性变化的。

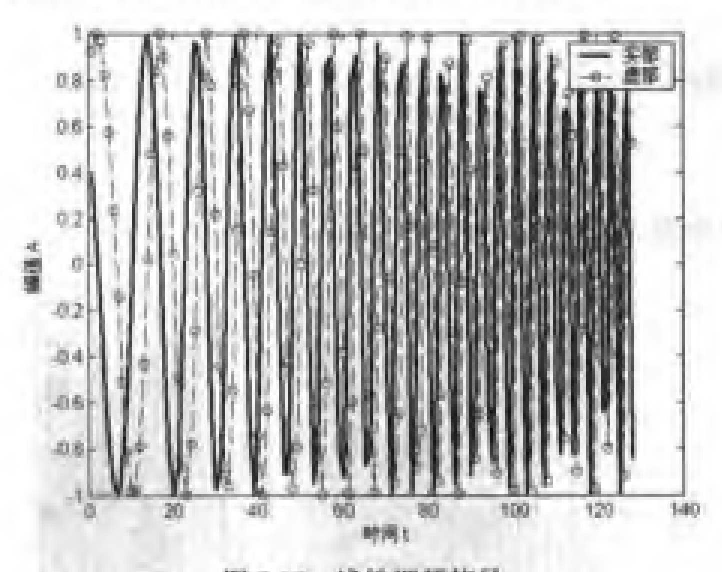


图 3.19 线性调频信号

4. fmodany 函数

【语法格式】

① [y,iflaw]= fmodany (iflaw)

② [y,iflaw]= fmodany (iflaw,t0)

【使用说明】生成任意频率的调频信号，其瞬时频率逼近于向量 iflaw。

输入参数：iflaw 是瞬时频率向量；t0 是时间参考点。

输出参数：y 返回产生的信号；iflaw 为瞬时频率。

例程 3-14：生成任意频率的调频信号。

f3_14.m

```
%产生线性信号
[sig1,ifl1]=fmlin(100);
%产生正弦调频信号
[sig2,ifl2]=fmsin(100);
%瞬时频率
iflaw=[ifl1;ifl2];
%产生信号
```

```
sig=fmodany(iflaw);
%显示
figure(1);
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');

figure(2);
plot(iflaw,'LineWidth',2);
xlabel('时间 t');
ylabel('频率 f');
```

计算得到的信号如图 3.20 所示，其瞬时频率如图 3.21 所示。

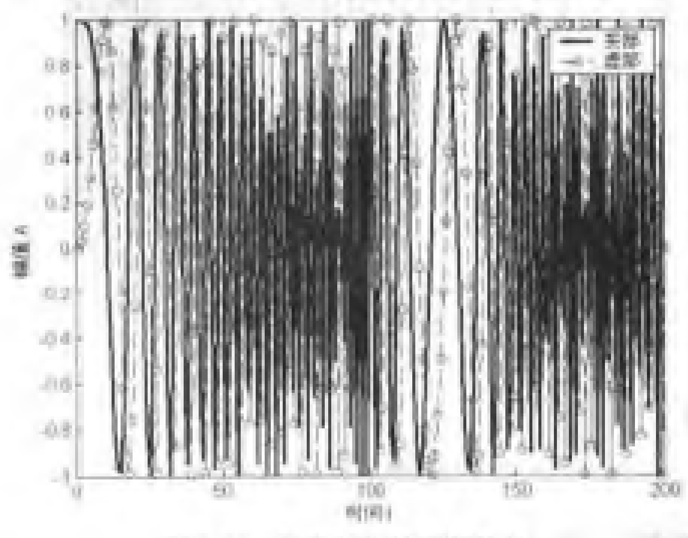


图 3.20 任意频率的调频信号

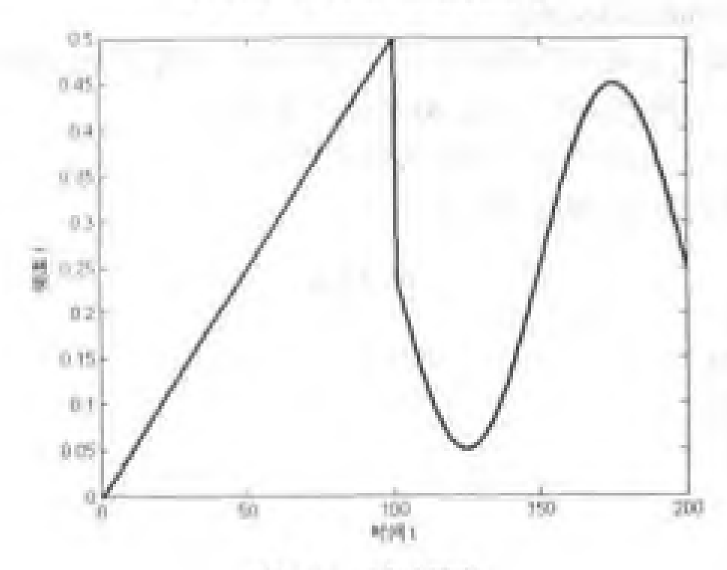


图 3.21 瞬时频率

5. fmpar 函数

【语法格式】

① [y,iflaw]= fmpar (N,P1)

② [y,iflaw]= fmpar (N,P1,P2,P3)

【使用说明】生成抛物线频率的调频信号，即

$$y(t)=\exp[j2\pi(a_0t+\frac{a_1}{2}t^2+\frac{a_2}{3}t^3)]$$

输入参数：N 是产生的信号点数；当输入参数个数等于 2 时，P1 是含有多项式瞬时相位三个系数的向量$[a_0,a_1,a_2]$；当输入参数个数等于 4 时，P1(P2,P3)是时频点$[t_i, f_i]$，且t_i的单位为秒，f_i为归一化的频率；P2,P3 同 P1。

输出参数：y 返回产生的信号；iflaw 为瞬时频率。

例程 3-15：生成抛物线频率的调频信号。

f3_15.m

```
%产生信号
[sig,iflaw]=fmpar(200,[1 0.4],[100 0.05],[200,0.4]);
%显示
figure(1);
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');
figure(2);
plot(iflaw,'LineWidth',2);
xlabel('时间 t');
ylabel('频率 f');
```

计算得到的信号如图 3.22 所示，其瞬时频率如图 3.23 所示，是一抛物线。

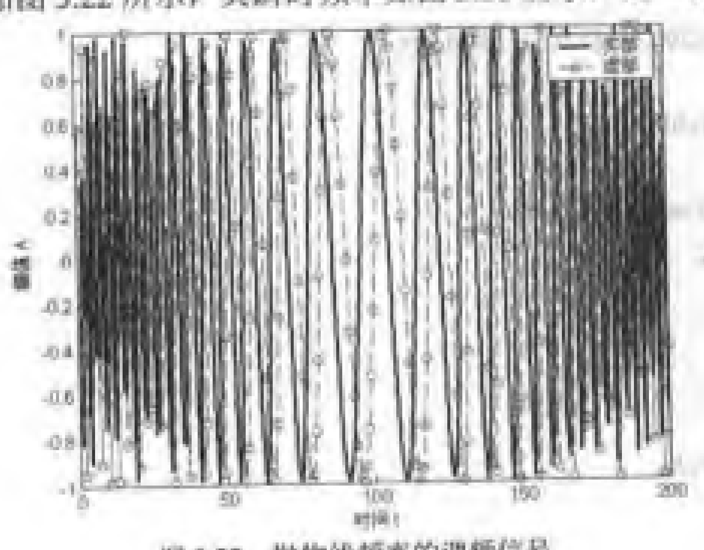


图 3.22 抛物线频率的调频信号

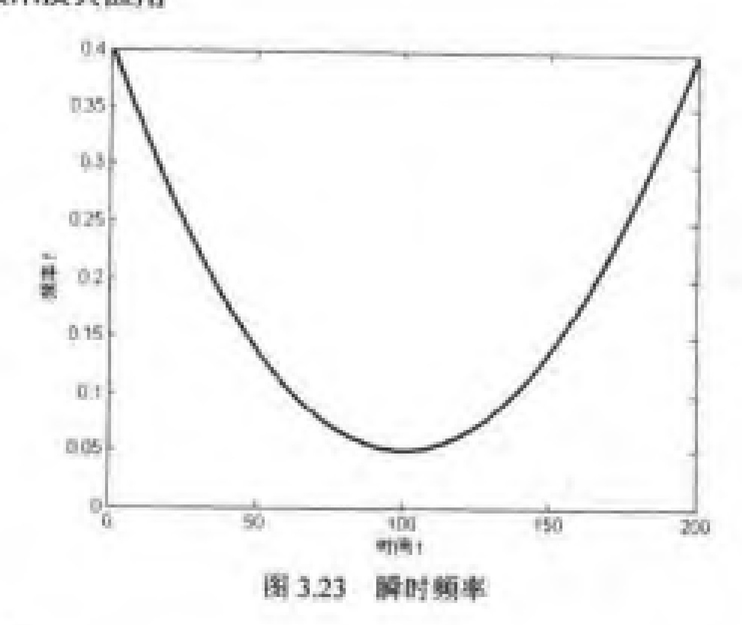


图 3.23　瞬时频率

6. fmpower 函数

【语法格式】

① [y,iflaw]= fmpower (N,k,P1)

② [y,iflaw]= fmpower (N, k,P1,P2)

【使用说明】生成幂指数频率的调频信号，即

$$y(t)=\exp[j2\pi(f_0t+\frac{c}{1-k}|t|^{1-k})]$$

输入参数：N 是产生的信号点数；k 是幂指数；当输入参数个数等于 3 时，P1 是含有幂指数瞬时频率两个系数的向量[$f0$, c]；当输入参数个数等于 4 时，P1(P2)是时频点[t_i, f_i]，且 t_i 的单位为秒，f_i 为归一化的频率；P2 同 P1。

输出参数：y 返回产生的信号；iflaw 为瞬时频率。

例程 3-16：生成幂指数频率的调频信号。

f3_16.m

```
%产生信号
[sig,iflaw]=fmpower(200,0.5,[1 0.5],[180 0.1]);
%显示
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');

figure(2);
plot(iflaw,'LineWidth',2);
xlabel('时间 t');
ylabel('频率 f');
```

计算得到的信号如图 3.24 示，其瞬时频率如图 3.25 所示，是一幂函数。

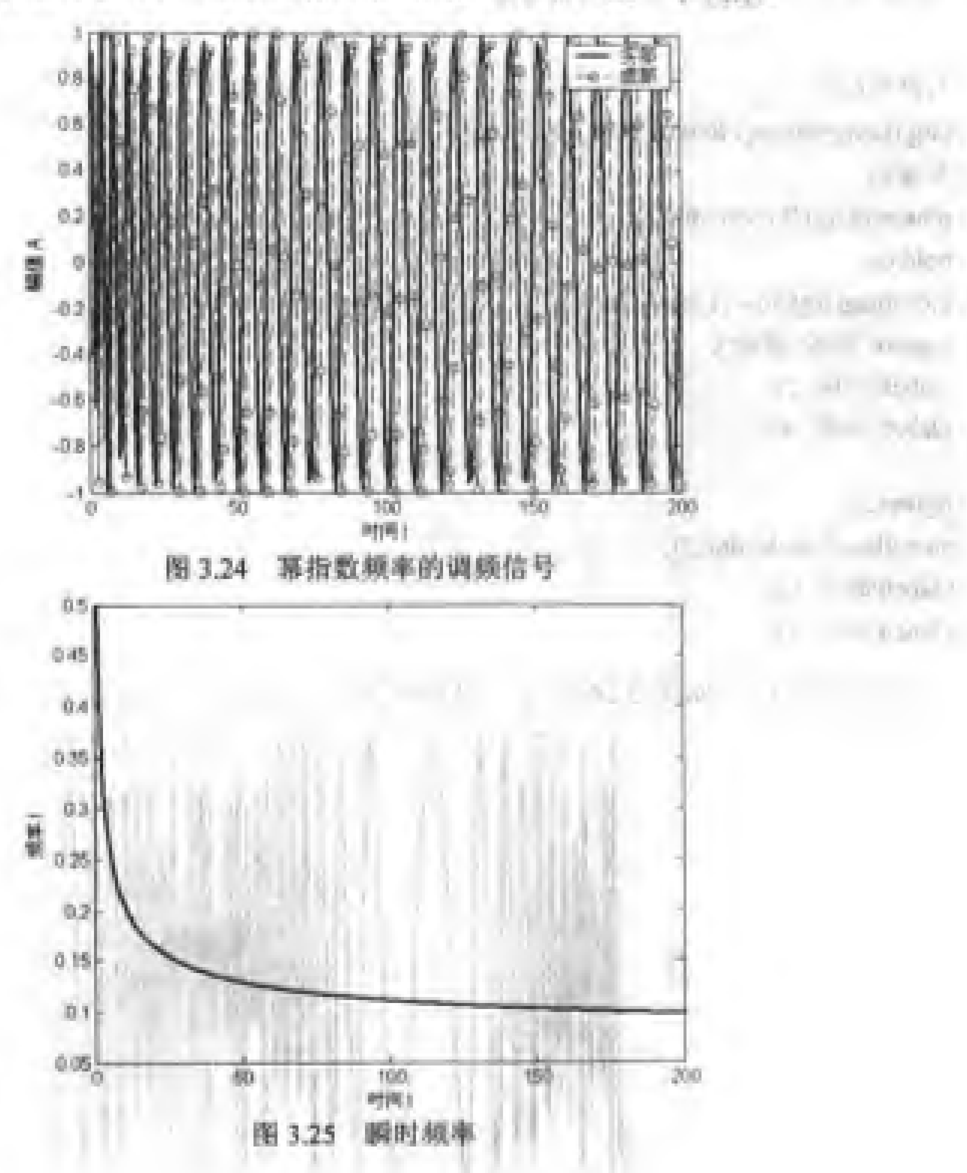


图 3.24　幂指数频率的调频信号

图 3.25　瞬时频率

7. fmsin 函数

【语法格式】

① [y,iflaw]= fmsin (N,fmin)

② [y,iflaw]= fmsin (N,fmin,fmax)

③ [y,iflaw]= fmsin (N,fmin,fmax,period)

④ [y,iflaw]= fmsin (N,fmin,fmax,period,t0)

⑤ [y,iflaw]= fmsin (N,fmin,fmax,period,t0,f0)

⑥ [y,iflaw]= fmsin (N,fmin,fmax,period,t0,f0,pm1)

【使用说明】生成正弦调频信号，其最小频率为 fmin，最大频率为 fmax。

输入参数：N 是产生的信号点数；fmin 是最小的归一化频率；fmax 是最大的归一化频率；period 是正弦调频信号的周期；t0 是相位的时间参考点；f0 是时刻 t0 时的归一化频率；pm1 是时刻 t0 时的频率方向。

输出参数：y 返回产生的信号；iflaw 为瞬时频率。

例程 3-17：生成正弦调频信号。

f3_17.m

```
%产生信号
[sig,iflaw]=fmsin(140,0.05,0.45,100,20,0.3,-1.0);
%显示
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');

figure(2);
plot(iflaw,'LineWidth',2);
xlabel('时间 t');
ylabel('频率 f');
```

计算得到的信号如图 3.26 所示，其瞬时频率如图 3.27 所示，是一正弦函数。

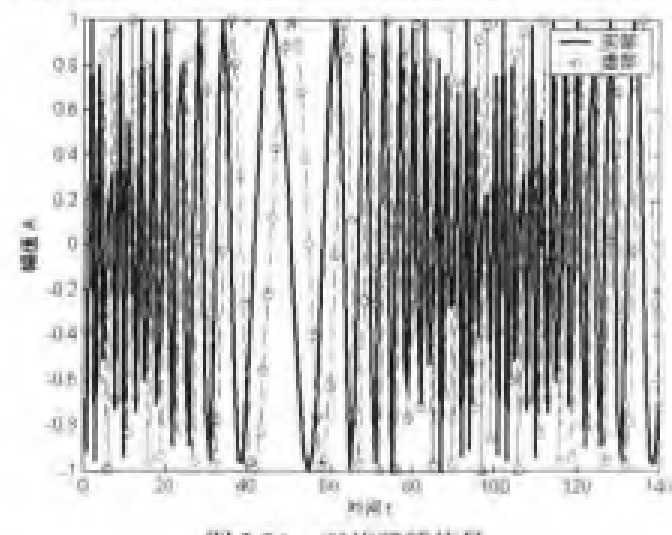


图 3.26 正弦调频信号

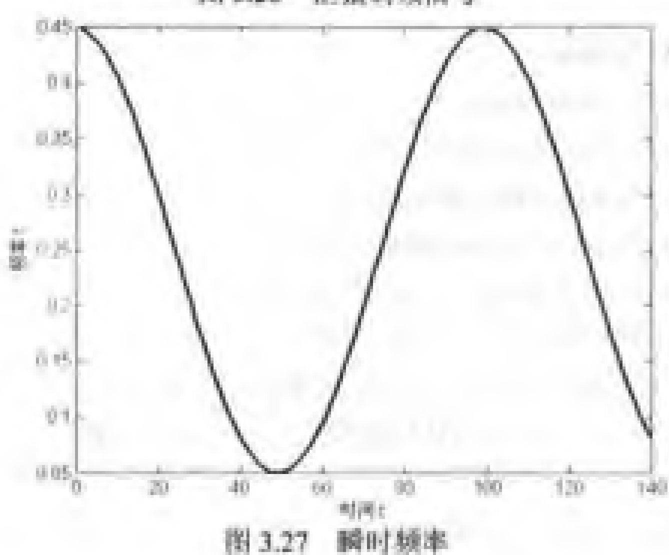


图 3.27 瞬时频率

（3）第三类信号函数

1. anaask 函数

【语法格式】

① [y,am]= anaask(N)

② [y,am]= anaask(N,ncomp)

③ [y,am]= anaask(N,ncomp,f0)

【使用说明】生成幅值键移信号。

输入参数：N 是点数；ncomp 是每个分量的点数；f0 是归一化的频率。

输出参数：y 返回生成的信号；am 返回得到的调制幅值。

例程 3-18：生成幅值键移信号。

f3_18.m

```
%产生信号
[sig,am]=anaask(512,64,0.05);
%显示
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');

figure(2);
plot(am,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
```

计算得到的信号如图 3.28 所示，其调制幅值如图 3.29 所示

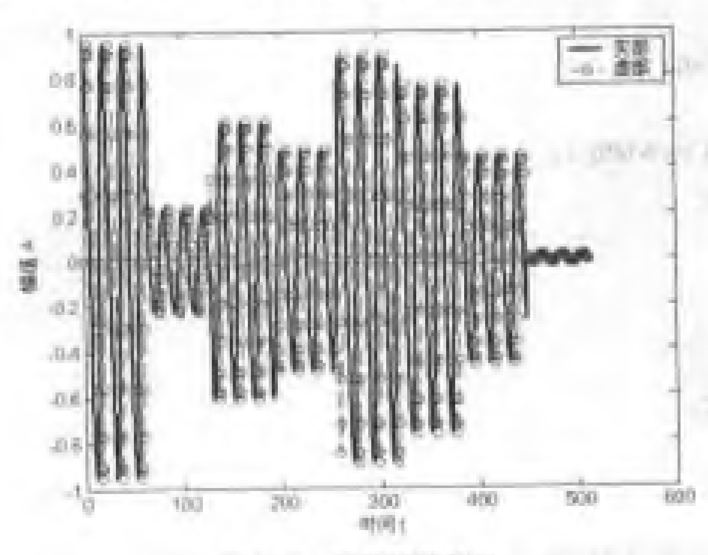


图 3.28　幅值键移信号

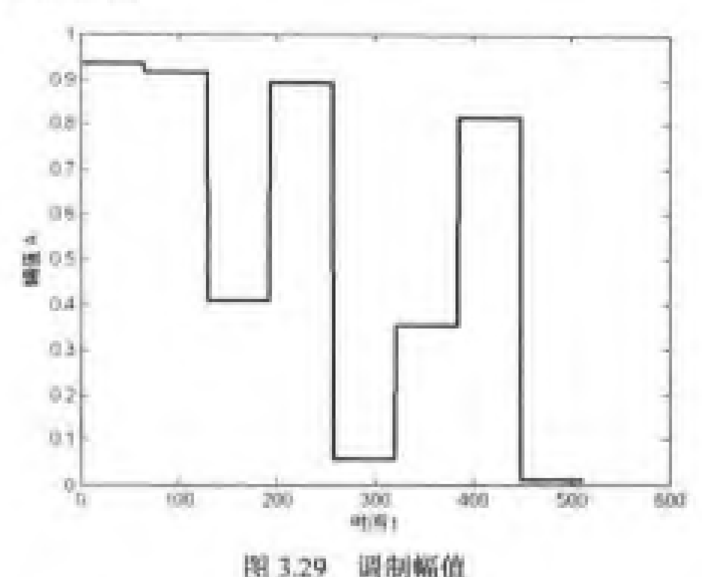


图 3.29　调制幅值

2. anabpsk 函数

【语法格式】

① [y,am]= anabpsk (N)

② [y,am]= anabpsk (N,ncomp)

③ [y,am]= anabpsk (N,ncomp,f0)

【使用说明】生成二进制相位键移信号。

输入参数：N 是点数；ncomp 是每个分量的点数；f0 是归一化的频率。

输出参数：y 返回生成的信号；am 返回得到的幅值包络。

例程 3-19：生成二进制相位键移信号。

f3_19.m

```
%产生信号
[sig,am]=anabpsk(300,30,0.1);
%显示
%实部和虚部
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');

figure(2);
plot(am,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
```

计算得到的信号如图 3.30 所示，其调制幅值如图 3.31 所示。

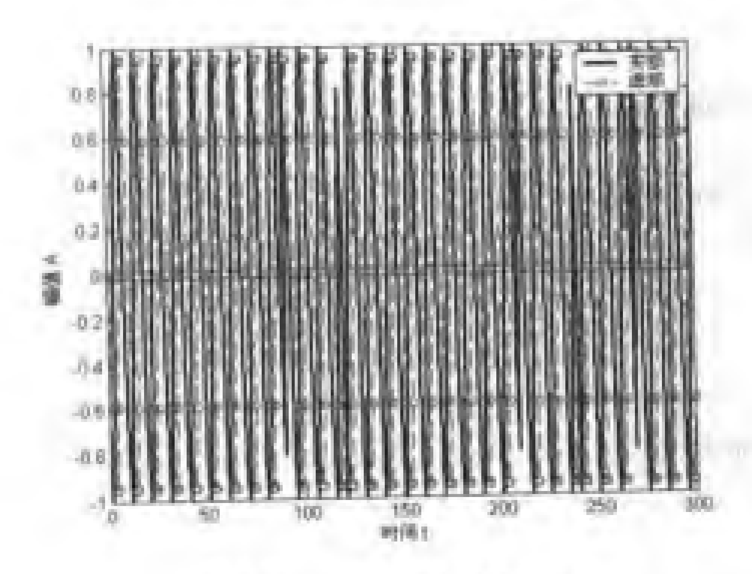


图 3.30 二进制相位键移信号

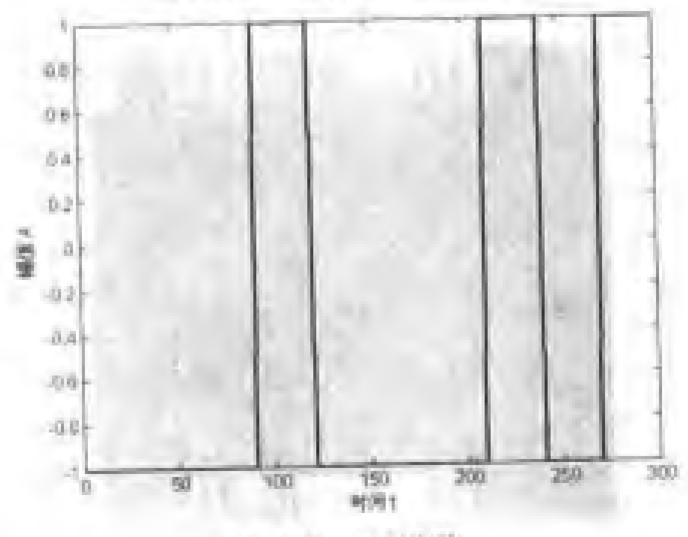


图 3.31 调制幅值

3. anafsk 函数

【语法格式】

① [y,iflaw]= anafsk(N)

② [y, iflaw]= anafsk(N,ncomp)

③ [y, iflaw]= anafsk(N,ncomp,nbf)

【使用说明】生成频率键移信号。

输入参数：N 是点数；ncomp 是每个分量的点数；nbf 是不同频率的个数。

输出参数：y 返回生成的信号；iflaw 返回瞬时频率。

例程 3-20：生成频率键移信号。

f3_20.m

```
%产生信号
[sig,iflaw]=anafsk(512,64,5);
```

```
%显示
%实部和虚部
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');

figure(2);
plot(iflaw,'LineWidth',2);
xlabel('时间 t');
ylabel('频率 f');
```

计算得到的信号如图 3.32 所示，其瞬时频率如图 3.33 所示。

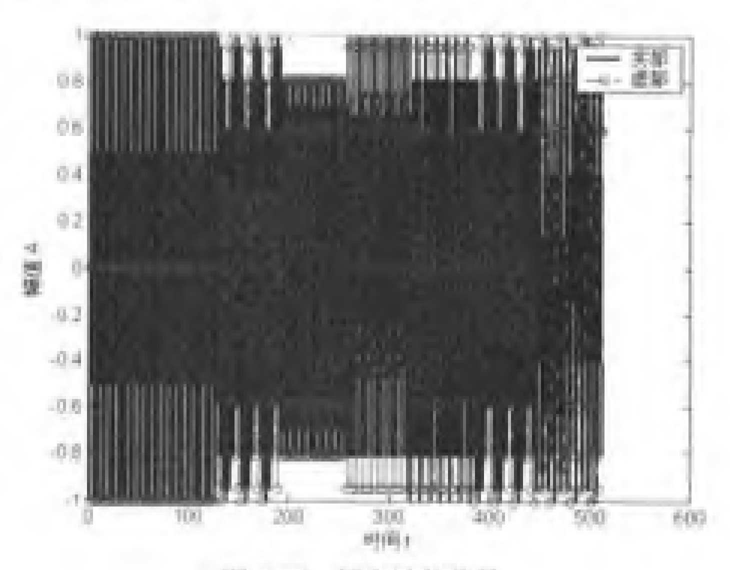


图 3.32　频率键移信号

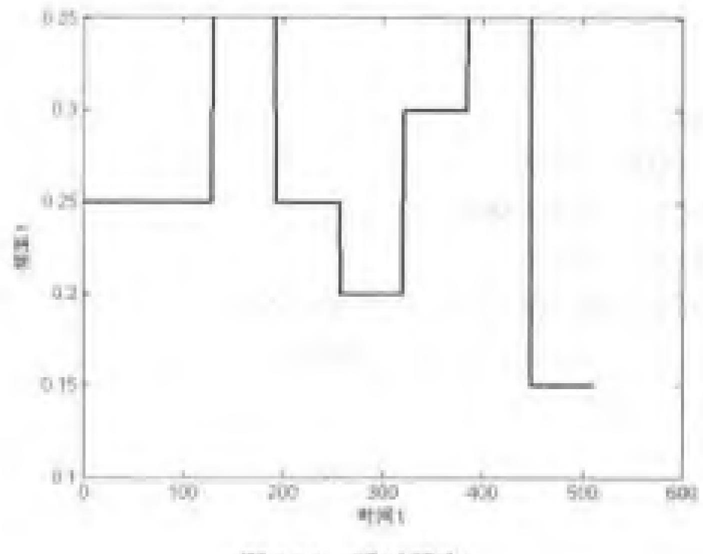


图 3.33　瞬时频率

4. anaqpsk 函数

【语法格式】

① [y,pm0]= anaqpsk(N)

② [y,pm0]= anaqpsk(N,ncomp)

③ [y,pm0]= anaqpsk(N,ncomp,f0)

【使用说明】生成四进制相位键移信号。

输入参数：N 是点数；ncomp 是每个分量的点数；f0 是归一化的频率。

输出参数：y 返回生成的信号；pm0 返回每个分量的初始相位。

例程 3-21：生成四进制相位键移信号。

f3_21.m

```
%产生信号
[sig,pm]=anaqpsk(512,64,0.05);
%显示
%实部和虚部
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--','LineWidth',1);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');

figure(2);
plot(pm,'LineWidth',2);
xlabel('时间 t');
ylabel('初始相位 phi');
```

计算得到的信号如图 3.34 所示，其初始相位如图 3.35 所示。

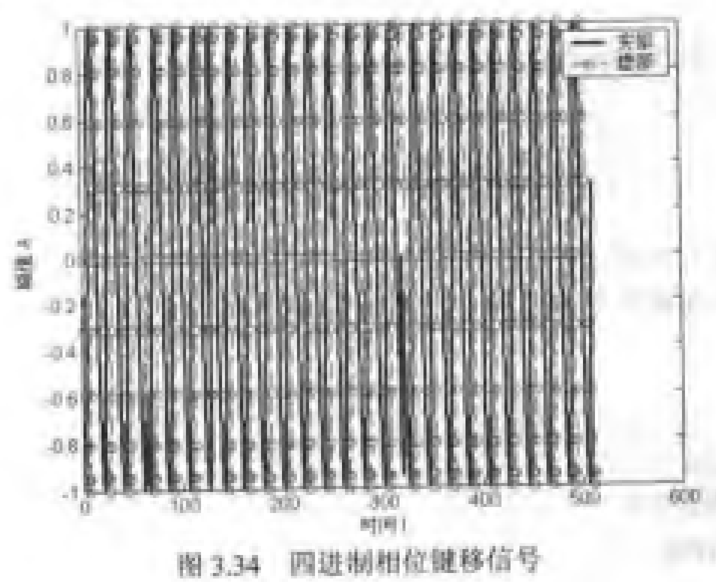


图 3.34　四进制相位键移信号

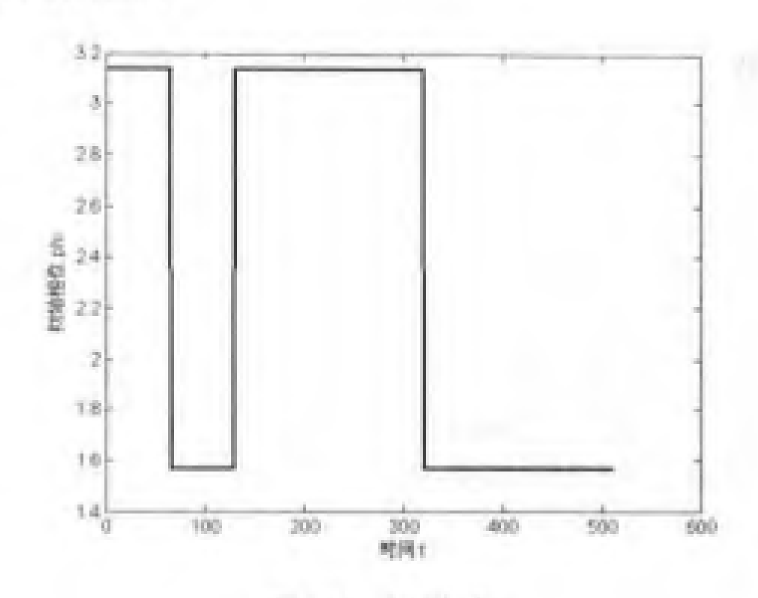


图 3.35　初始相位

3.2.2　非平稳信号生成实例

例程 3-22：线性调频信号乘以高斯幅值调制信号。

f3_22.m

```
%线性调频信号
fm1=fmlin(256,0,0.5);
%高斯幅值调制信号
am1=amgauss(256);
%合成
sig=am1.*fm1;
%显示
%实部
plot(real(sig),'LineWidth',2);
hold on;
%虚部
plot(imag(sig),'ro--','LineWidth',2);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');
```

计算的结果如图 3.36 所示。

例程 3-23：常数调频信号乘以单边指数窗幅值调制信号。

f3_23.m

```
%常数调频信号
fm1=fmconst(256,0.2);
%单边指数窗幅值调制信号
am1=amexpo1s(256,100);
```

```
%合成
sig=am1.*fm1;
%显示
%实部
plot(real(sig),'LineWidth',2);
hold on;
%虚部
plot(imag(sig),'ro--','LineWidth',2);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');
```

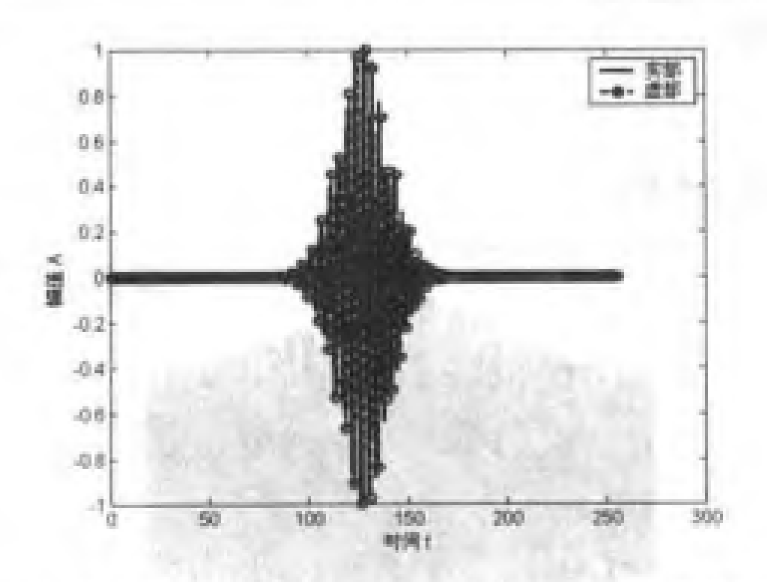


图 3.36　单分量非平稳信号波形（线性调频信号乘以高斯调幅信号）

计算的结果如图 3.37 所示。

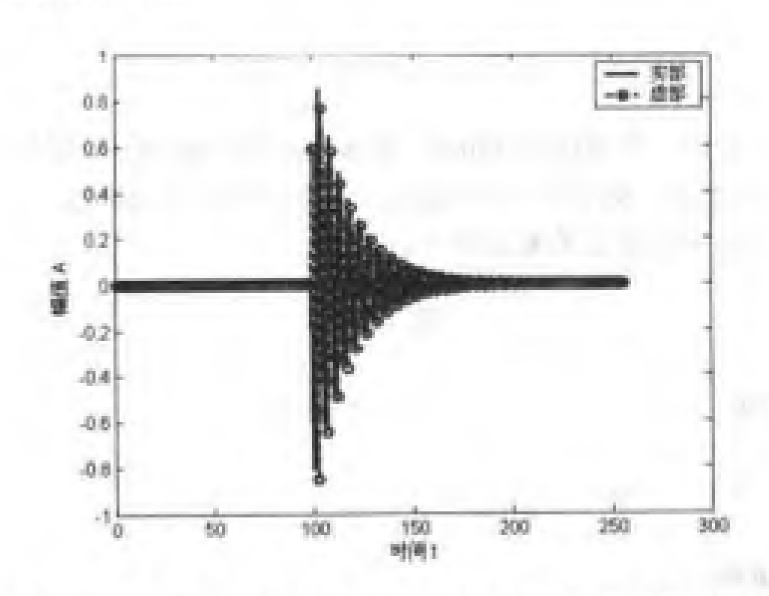


图 3.37　单分量非平稳信号波形（常调频信号乘以指数调幅信号）

例程 3-24：多普勒信号（利用固定观测器测量到的运动目标的纯频信号）。

f3_24.m

```
%多普勒信号
[fm,am]=doppler(256,200,4000/60,10,50);
sig=am.*fm;
%显示
%实部
plot(real(sig),'LineWidth',2);
hold on;
%虚部
plot(imag(sig),'ro--','LineWidth',2);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');
```

计算的结果如图 3.38 所示。

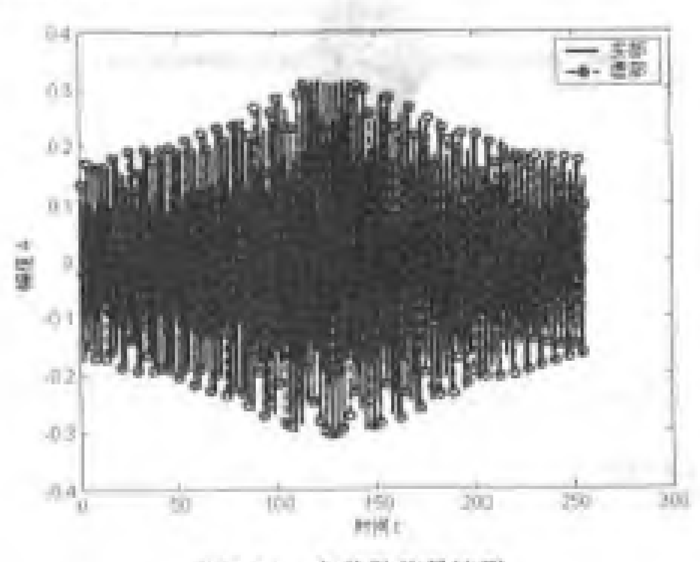


图 3.38　多普勒信号波形

上述信号实际上对应一个速度为 50m/s，距离观测器 10m 的运动目标信号，比如汽车，4000 对应其发动机的转速，观测器（比如雷达）的采样频率为 200Hz。

例程 3-25：产生不同信噪比的暂态信号。

f3_25.m

```
%暂态信号
fm1=fmlin(256,0,0.5);
am1=amgauss(256);
sig=am1.*fm1;
%噪声
noise=noisecg(256,0.8);
%设置信噪比
```

```
dB=-10;
sign=sigmerge(sig,noise,dB);
%显示
%实部
plot(real(sign),'LineWidth',2);
hold on;
%虚部
plot(imag(sign),'ro--','LineWidth',2);
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');
```

计算的结果如图 3.39 所示。

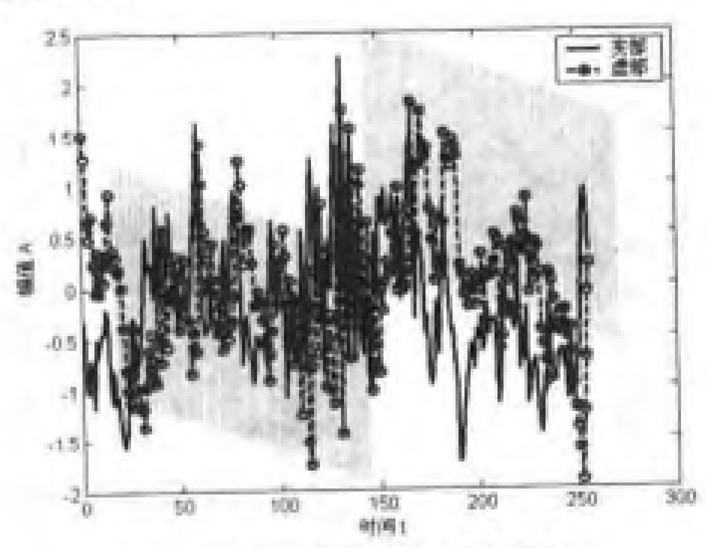


图 3.39　信噪比为-10dB 的高斯暂态信号

以上产生的非平稳信号都是单分量信号，即在任一时刻，信号只存在单一频率分量。实际中信号往往在同一时刻具有多个频率分量，即多分量信号。时频分析工具箱也可以用于产生这样的非平稳信号。

例程 3-26：两个线性调频信号的叠加。

f3_26.m

```
%样本点数
N=256;
%线性调频信号
x1=fmlin(N,0,0.2);
x2=fmlin(N,0.3,0.5);
%叠加
x=x1+x2;
%瞬时频率
ifr=instfreq(x);
%群延迟
```

```
f=0:0.01:0.5;
gd=sgrpdlay(x,f);
%显示
figure(1)
plot(ifr,'LineWidth',2);
xlabel('时间 t');
ylabel('归一化的频率 f');
figure(2)
plot(gd,f,'LineWidth',2);
xlabel('时间 t');
ylabel('归一化的频率 f');
```

计算的结果如图 3.40、图 3.41 所示。

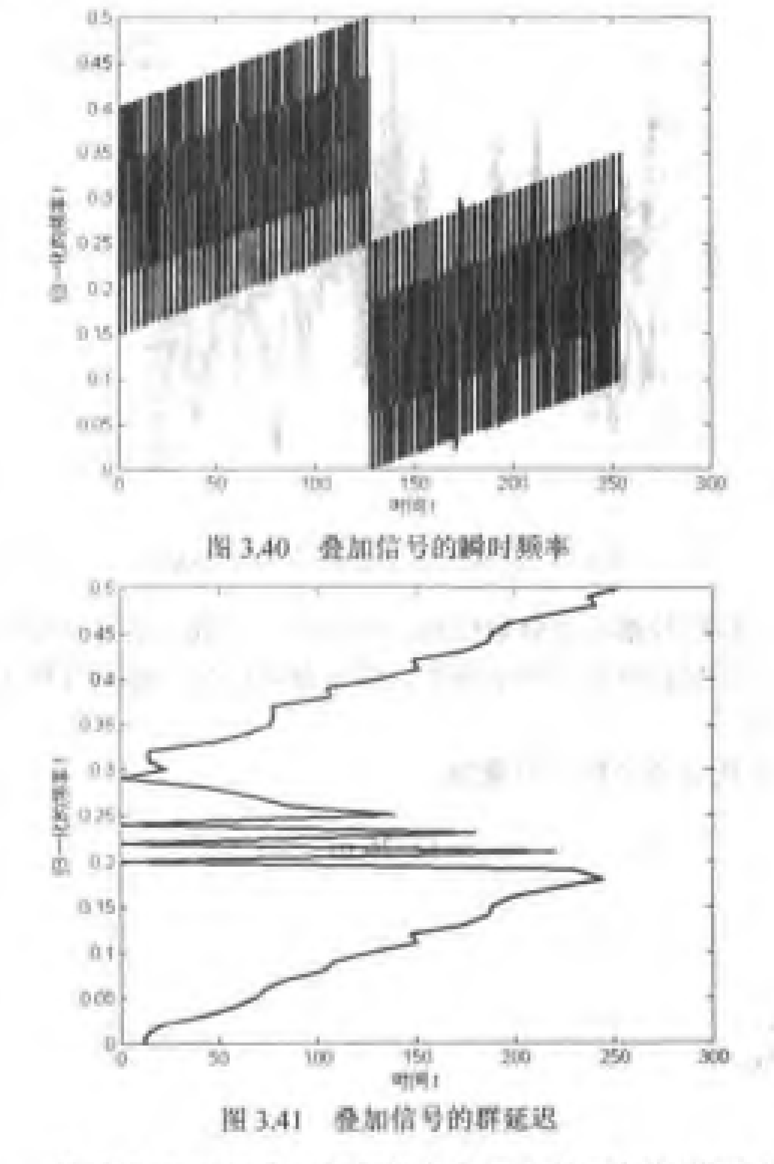


图 3.40　叠加信号的瞬时频率

图 3.41　叠加信号的群延迟

在每个时刻 t，一个理想的时频表示应该能描述具有相同幅值的两个不同的频率分量，但是从图 3.40 和图 3.41 可以看出，瞬时频率和群延迟还不能充分揭示所有非平稳信号的信息，因此进一步需要在时间和频率二维空间来联合表示非平稳信号。

第 4 章　线性时频表示

正如前文所说，傅立叶变换不能分析非平稳信号，而瞬时频率和群延迟也不能完全揭示多分量非平稳信号的信息，因此需要进行联合时频分析。时频分析的第一类就是线性时频表示，以短时傅立叶变换和小波变换为代表。

本章主要内容包括：

- 短时傅立叶变换；
- 时间-尺度分析。

4.1　短时傅立叶变换

4.1.1　相关的函数基本用法

短时傅立叶变换定义为：

$$\mathrm{STFT}_z(t,f)=\int_{-\infty}^{\infty} z(t')\eta^*(t'-t)e^{-j2\pi ft'}dt'$$

短时傅立叶变换的时间分辨率与分析窗函数$\eta(t)$的时间域宽度成正比，而其频率分辨率与分析窗$\eta(t)$的频宽成正比。于是，一个好的时间分辨率需要一个短的窗函数$\eta(t)$，同时一个高的频率分辨率需要一个窄带的滤波器，也即一个长的窗函数$\eta(t)$，因此实际应用时应该在这两个指标之间取得一种平衡。

时频分析工具箱中计算短时傅立叶变换的函数为 tfrstft.m.

1. tfrstft 函数

【语法格式】

① [tfr,t,f]=tfrstft(x)

② [tfr,t,f]=tfrstft(x,t)

③ [tfr,t,f]=tfrstft(x,t,N)

④ [tfr,t,f]=tfrstft(x,t,N,h)

⑤ [tfr,t,f]=tfrstft(x,t,N,h,trace)

【使用说明】tfrstft 计算离散信号 x 的短时傅立叶变换。

输入参数：N 是频率点的数目 N=length(x)；*h* 表示频率平滑函数；如果 trace≠0，则跟踪算法过程。

输出参数：tft 返回信号的时频值（为复换）；t 是返回的时间坐标（1:*length*(x))；f 是返回的频率坐标（归一化），从–0.5 到 0.5；特别地，当没有输出参数时，函数 tfrstft 会调用函数 tfrqview，显示短时傅立叶变换模的平方值。

例程 4-1：计算信号的短时傅立叶变换。

f4_1.m

```
%产生信号
sig=[fmlin(128,0.05,0.45);fmlin(128,0.35,0.15)];
%时域波形
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--');
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');
%计算短时傅立叶变换
[tfr,t,f]=tfrstft(sig);
%幅值图
figure(2)
plot(t,abs(tfr(1:128,:)),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
%相位图
figure(3)
plot(t,angle(tfr(1:128,:)),'LineWidth',2);
xlabel('时间 t');
ylabel('相位 phi');
%时频表示
figure(4)
contour(t,f(1:256),abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
```

产生的非平稳信号时域波形如图 4.1 所示，其短时傅立叶变换的幅值如图 4.2 所示，相位如图 4.3 所示，短时傅立叶变换的时频表示如图 4.4 所示。

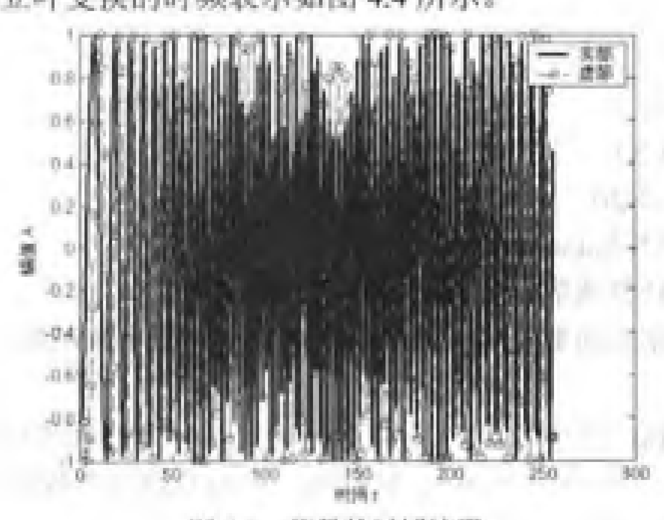


图 4.1　信号的时域波形

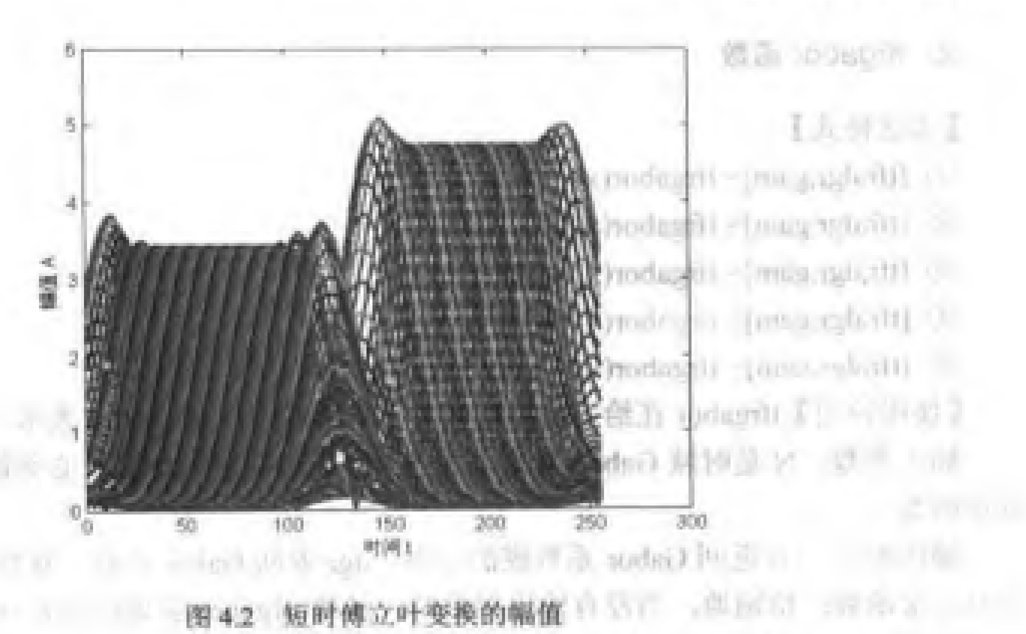


图 4.2 短时傅立叶变换的幅值

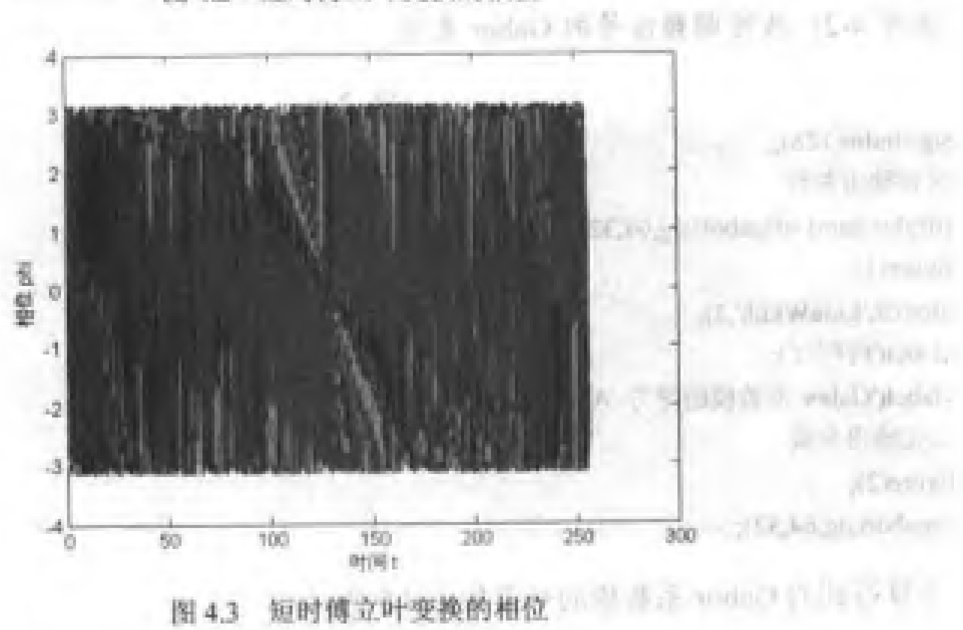


图 4.3 短时傅立叶变换的相位

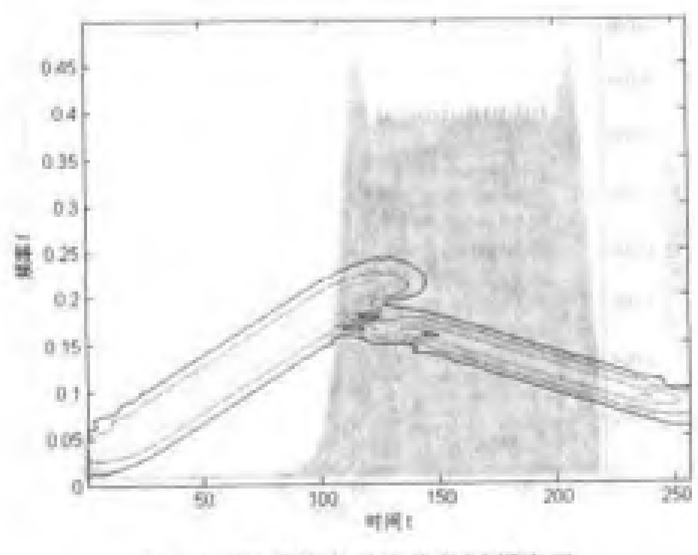


图 4.4 短时傅立叶变换的时频表示

2. tfrgabor 函数

【语法格式】

① [tfr,dgr,gam]= tfrgabor(x)

② [tfr,dgr,gam]= tfrgabor(x,N)

③ [tfr,dgr,gam]= tfrgabor(x,N,Q)

④ [tfr,dgr,gam]= tfrgabor(x,N,Q,h)

⑤ [tfr,dgr,gam]= tfrgabor(x,N,Q,h,trace)

【使用说明】tfrgabor 在给定的合成窗 h 下计算信号 x 的 Gabor 表示。

输入参数：N 是时域 Gabor 系数的数目；Q 是过采样的次数，必须能被 N 整除；h 是合成窗函数。

输出参数：tfr 返回 Gabor 系数模的平方；dgr 返回 Gabor 系数（复数）；gam 是与 h 相关的双正交函数；特别地，当没有输出参数时，函数 tfrgabor 会调用函数 tfrqview。

例程 4-2：线性调频信号的 Gabor 表示。

f4_2.m

```
sig=fmlin(128);
%有输出参数
[tfr,dgr,gam]=tfrgabor(sig,64,32);
figure(1);
plot(tfr,'LineWidth',2);
xlabel('时间 t');
ylabel('Gabor 系数模的平方 A');
%无输出参数
figure(2);
tfrgabor(sig,64,32);
```

计算得到的 Gabor 系数模的平方如图 4.5 所示。

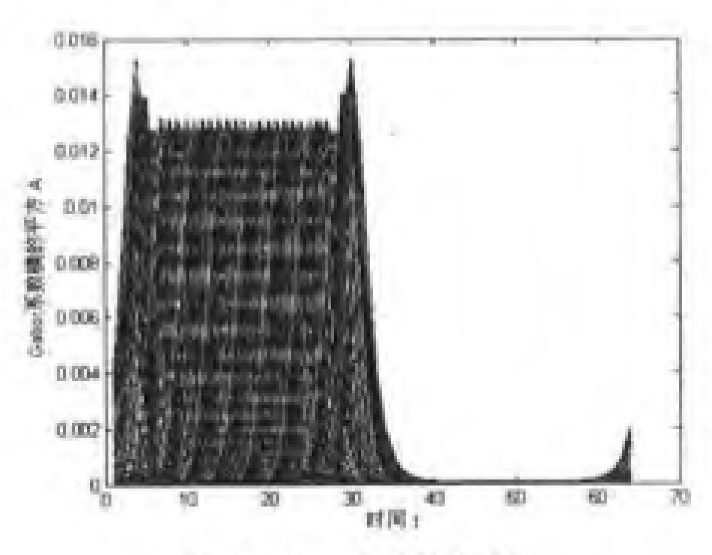


图 4.5　Gabor 系数模的平方

Gabor 变换的时频表示如图 4.6 所示。

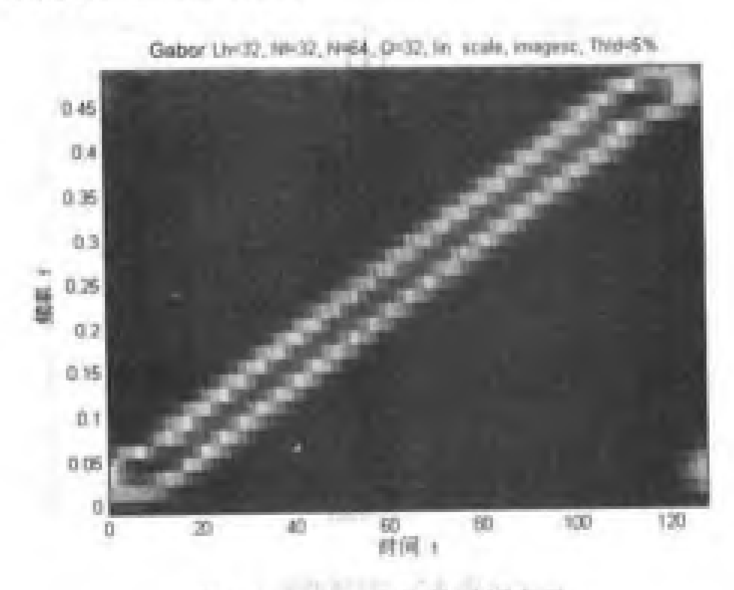


图 4.6　Gabor 变换的时频表示

4.1.2　实例分析

例程 4-3：在短时傅立叶变换中设置理想的时间分辨率，即窗函数$\eta(t)$选择为δ函数，此时信号的短时傅立叶变换在时间轴上完全局部化，而没有任何频率分辨率。

f4_3.m

```
%产生非平稳信号
sig=real(amgauss(128).*fmlin(128));
%%时域波形
figure(1)
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
%设置窗函数
h=1;
%计算短时傅立叶变换
sig=hilbert(sig);
[tfr,t,f]=tfrstft(sig,1:128,128,h);

%时频表示
figure(2)
contour(t,f(1:length(f)/2),abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
```

产生的非平稳信号时域波形如图 4.7 所示，其短时傅立叶变换（窗函数为δ函数）如图 4.8 所示。

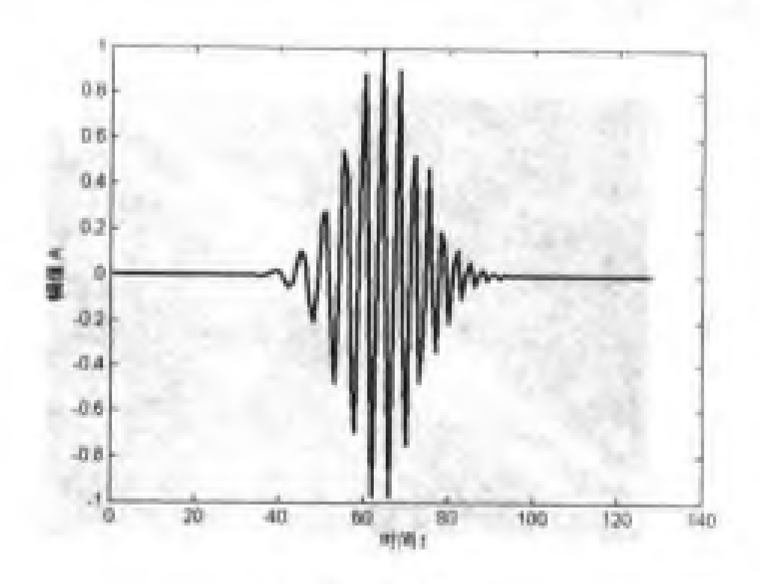


图 4.7　时域波形

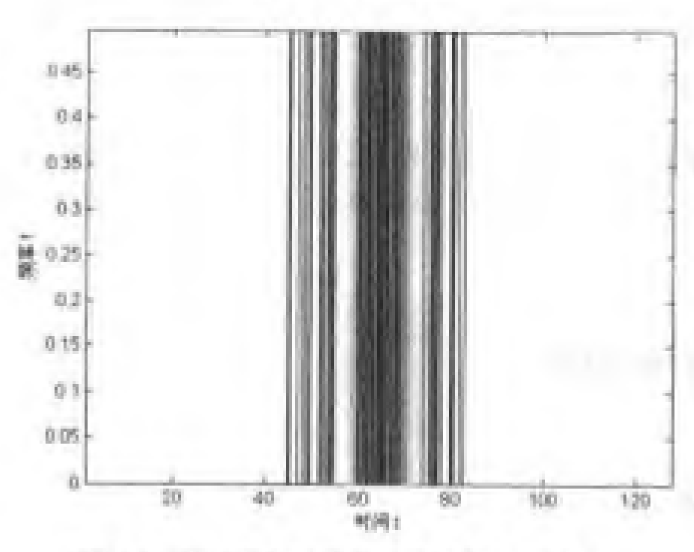


图 4.8　短时傅立叶变换（窗函数为δ函数）

例程 4-4：在短时傅立叶变换中设置理想的频率分辨率，即窗函数$\eta(t)$选择为常数函数，此时信号的短时傅立叶变换在频率轴上完全局部化，而没有任何时间分辨率。

f4_4.m

```
%产生非平稳信号
sig=real(amgauss(128).*fmlin(128));
%%时域波形
figure(1)
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
%设置窗函数
h=ones(127,1);
```

```
%计算短时傅立叶变换
sig=hilbert(sig);
[tfr,t,f]=tfrstft(sig,1:128,128,h);

%时频表示
figure(2)
contour(t,2*f(1:length(f)/2),abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
```

窗函数为常数函数时，非平稳信号的短时傅立叶变换如图4.9所示。

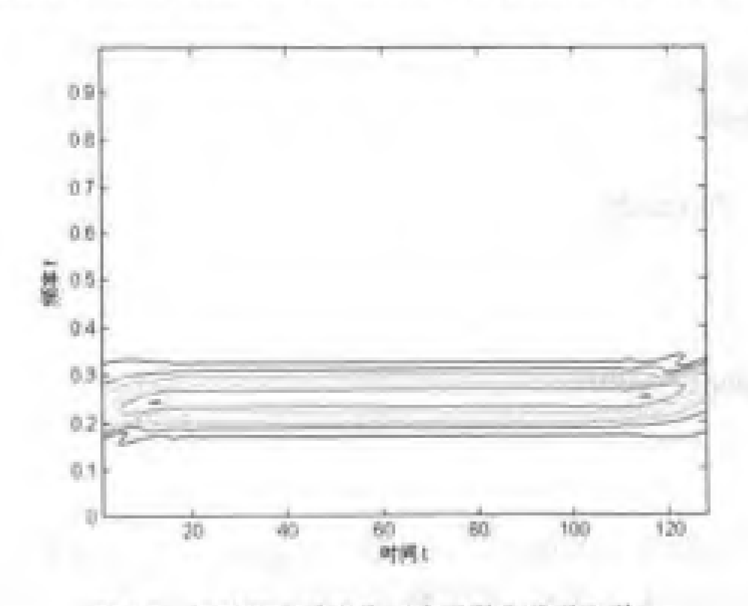


图4.9　短时傅立叶变换（窗函数为常数函数）

例程4-5：为了说明分析窗函数$\eta(t)$的形状和长度对短时傅立叶变换的影响，考虑计算由两个具有不同位置的暂态信号组成的非平稳信号的短时傅立叶变换。

f4_5.m

```
%产生非平稳信号
%%暂态信号1
sig1=real(amgauss(128,45).*fmconst(128,0.25,45));
%%暂态信号2
sig2=real(amgauss(128,85).*fmconst(128,0.25,85));
sig=sig1+sig2;
%时域波形
figure(1)
plot(sig,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');

%设置窗函数1
```

```
h1=window(65,'hamming');
%计算短时傅立叶变换
sig=hilbert(sig);
[tfr,t,f]=tfrstft(sig,1:128,128,h1);

%时频表示
figure(2)
contour(t,2*f(1:length(f)/2),abs(tfr));
xlabel('时间 t');
ylabel('频率 f');

%设置窗函数 2
h2=window(17,'hamming');
%计算短时傅立叶变换
sig=hilbert(sig);
[tfr,t,f]=tfrstft(sig,1:128,128,h2);

%时频表示
figure(3)
contour(t,2*f(1:length(f)/2),abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
```

生成的非平稳信号如图 4.10 所示，它由两个分量组成。当窗函数为 65 点的 hamming 窗时的短时傅立叶变换如图 4.11 所示，此时频率分辨率较好，但是在时间轴上不能区分信号中的两个分量；而当窗函数为 17 点的 hamming 窗时的短时傅立叶变换如图 4.12 所示，此时降低了频率分辨率，不过在时间轴上能够区分信号中的两个分量。

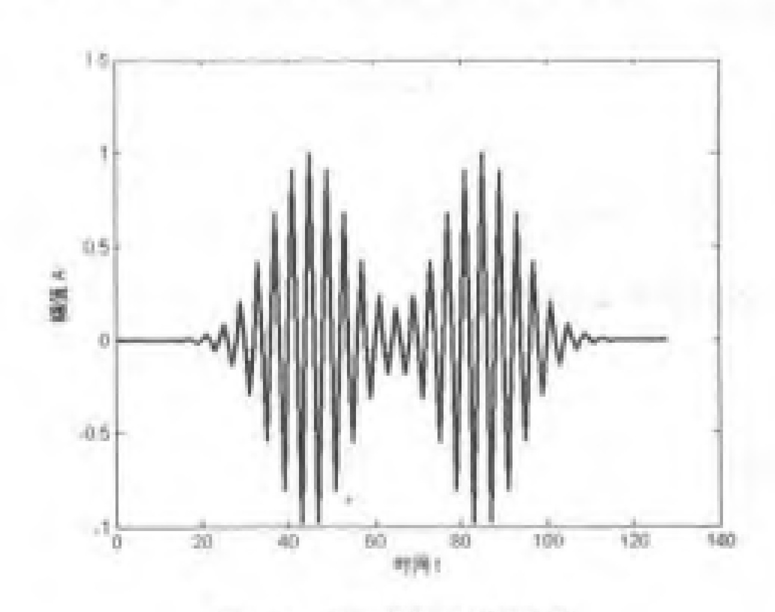


图 4.10　非平稳信号时域波形

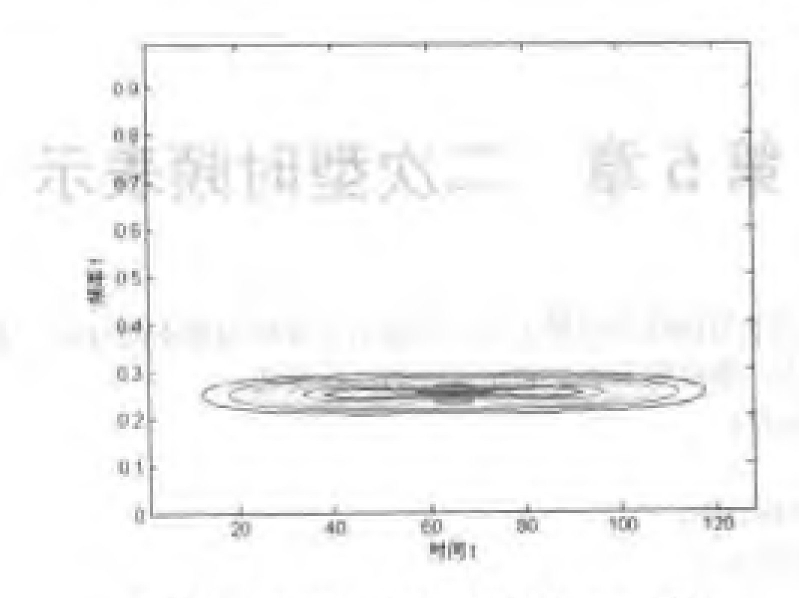


图 4.11 短时傅立叶变换（65 点 hamming 窗）

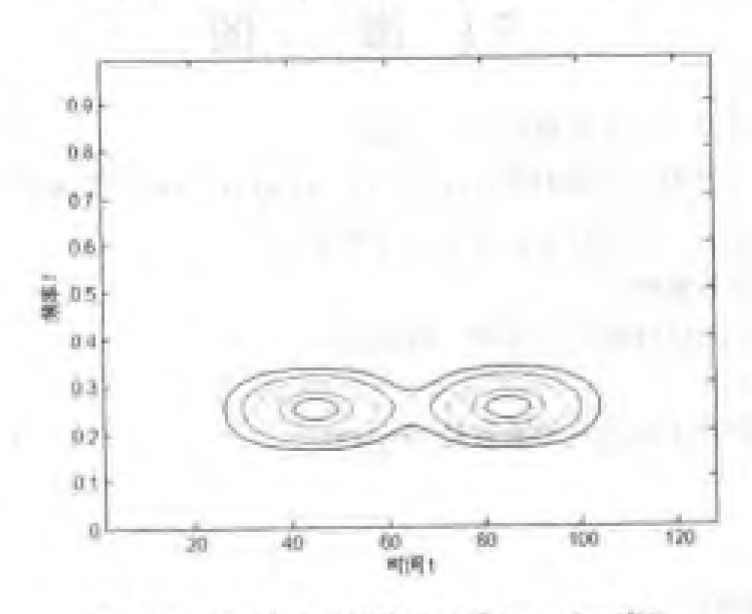


图 4.12 短时傅立叶变换（17 点 hamming 窗）

4.2 时间-尺度分析

常用的时间-尺度分析方法是小波变换，本书第 3 部分将对此进行详细描述，在此不做展开。它与短时傅立叶变换的区别在于：当尺度参数 a 变换时，小波的时宽和频宽同时发生变换，而形状保持不变，因此与短时傅立叶变换使用单一的分析窗不同，小波变换在高频处使用短窗函数，在低频处使用长窗函数。

第 5 章　二次型时频表示

第 4 章介绍的是信号的线性时频表示，即将信号分解为基本的分量。本章将讨论信号在时频面上的能量分布，即时频分布，它是信号的二次变换。

本章主要内容包括：

- 谱图；
- Cohen 类时频分布；
- Affine 类时频分布。

5.1 谱　　图

谱图定义为短时傅立叶变换模的平方，即

$$S_z(t,f)=|STFT_z(t,f)|^2=|\int_{-\infty}^{\infty} z(t')\eta^*(t'-t)e^{-j2\pi ft'}dt'|^2$$

它是实值、非负的二次型分布，具有以下性质：

（1）时移和频移不变性；

（2）时频分辨率同短时傅立叶变换一样受限；

（3）存在干扰项；

时频分析工具箱中计算谱图的函数是 tfrsp.m。

tfrsp 函数

【语法格式】

① [tfr,t,f]= tfrsp(x)

② [tfr,t,f]= tfrsp(x,t)

③ [tfr,t,f]= tfrsp(x,t,N)

④ [tfr,t,f]= tfrsp(x,t,N,h)

⑤ [tfr,t,f]= tfrsp(x,t,N,h,trace)

【使用说明】计算离散信号 *x* 的谱图。

输入参数：x 是离散的解析信号；N 是频率点数；h 是频域平滑窗函数。

输出参数：tfr 返回谱图；t 返回时间坐标；f 返回频率坐标。

例程 5-1：计算两个距离较近的线性调频信号叠加后的谱图。

f5_1.m

```
%产生非平稳信号
%%暂态信号 1
sig1=fmlin(128,0,0.4);
%%暂态信号 2
```

```
sig2=fmlin(128,0.1,0.45);
sig=sig1+sig2;
%时域波形
figure(1)
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--');
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');

%设置窗函数 1
h1=window(23,'gauss');
%计算谱图
[tfr,t,f]=tfrsp(sig,1:128,128,h1);
%时频分布
figure(2)
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
%设置窗函数 2
h2=window(63,'gauss');
%计算谱图
[tfr,t,f]=tfrsp(sig,1:128,128,h2);
%时频分布
figure(3)
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
```

由两个距离较近的线性调频信号叠加构成的非平稳信号波形如图 5.1 所示。

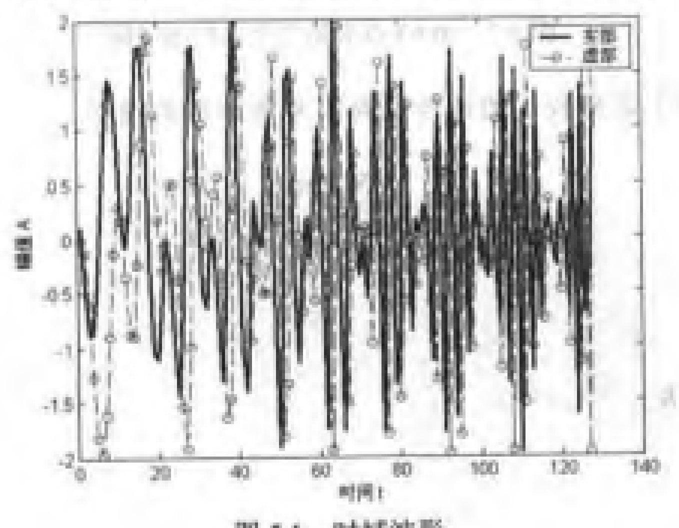


图 5.1 时域波形

当频域平滑窗取长度为 23 的高斯窗时，计算得到的谱图如图 5.2 所示；当频域平滑窗取长度为 63 的高斯窗时，计算得到的谱图如图 5.3 所示。比较图 5.2 和图 5.3 可知，由于两个线性调频信号相隔较近，所以不管取长还是短的频域平滑窗，谱图中的交叉项都会存在。

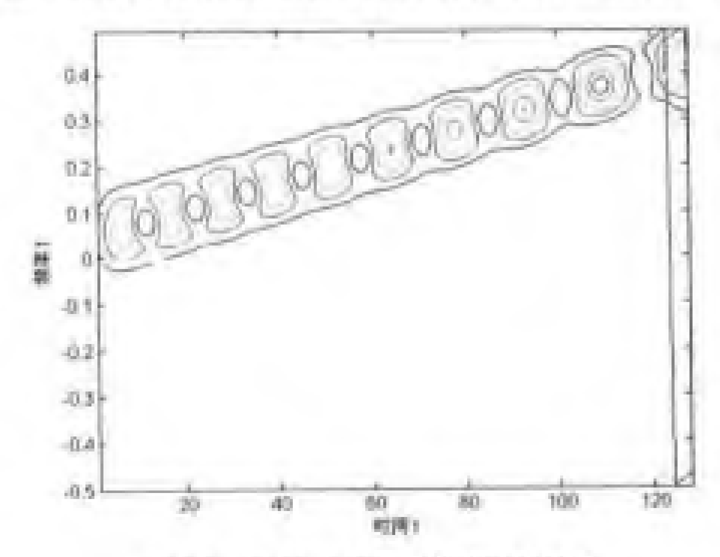


图 5.2　加 23 点高斯平滑窗时的谱图

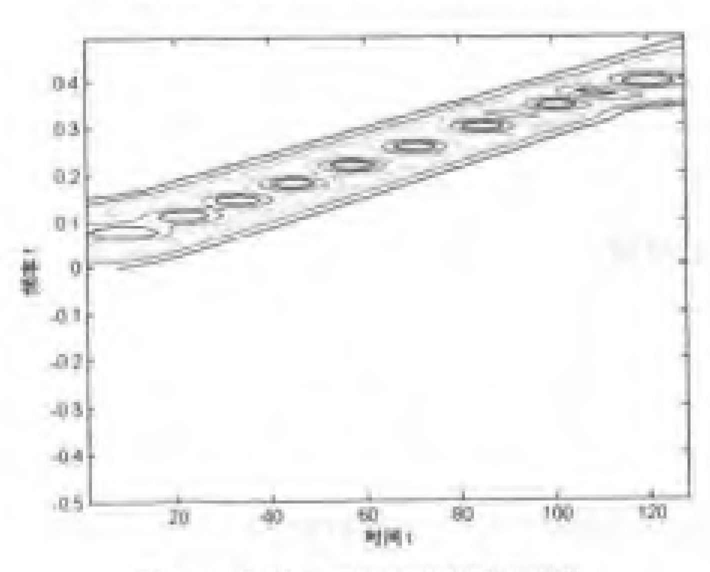


图 5.3　加 63 点高斯平滑窗时的谱图

例程 5-2：计算两个距离较远的线性调频信号叠加后的谱图。

f5_2.m

```
%产生非平稳信号
%%暂态信号 1
sig1=fmlin(128,0,0.3);
%%暂态信号 2
sig2=fmlin(128,0.2,0.4);
sig=sig1+sig2;
%时域波形
figure(1)
```

```
plot(real(sig),'LineWidth',2);
hold on;
plot(imag(sig),'ro--');
legend('实部','虚部');
xlabel('时间 t');
ylabel('幅值 A');

%设置窗函数 1
h1=window(23,'gauss');
%计算谱图
[tfr,t,f]=tfrsp(sig,1:128,128,h1);
%时频分布
figure(2)
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');

%设置窗函数 2
h2=window(63,'gauss');
%计算谱图
[tfr,t,f]=tfrsp(sig,1:128,128,h2);
%时频分布
figure(3)
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
```

由两个距离较远的线性调频信号叠加构成的非平稳信号波形如图 5.4 所示。

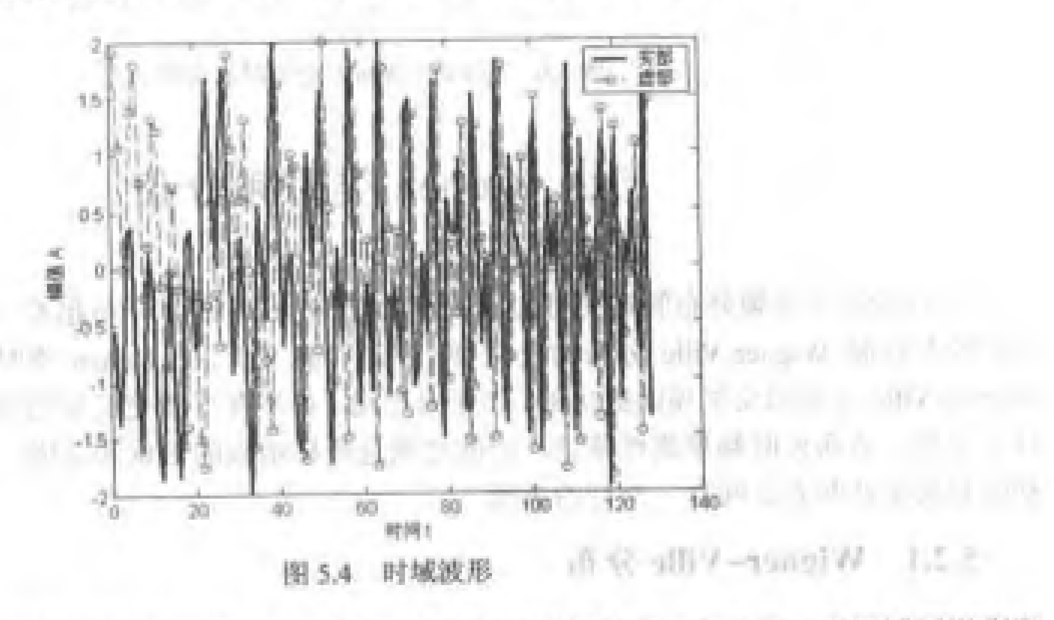


图 5.4 时域波形

当频域平滑窗取长度为 23 的高斯窗时，计算得到的谱图如图 5.5 所示；当频域平滑窗取

长度为 63 的高斯窗时，计算得到的谱图如图 5.6 所示。比较图 5.5 和图 5.6 可知，由于两个线性调频信号相隔较远，所以取长的频域平滑窗，可以消除谱图中的交叉项。

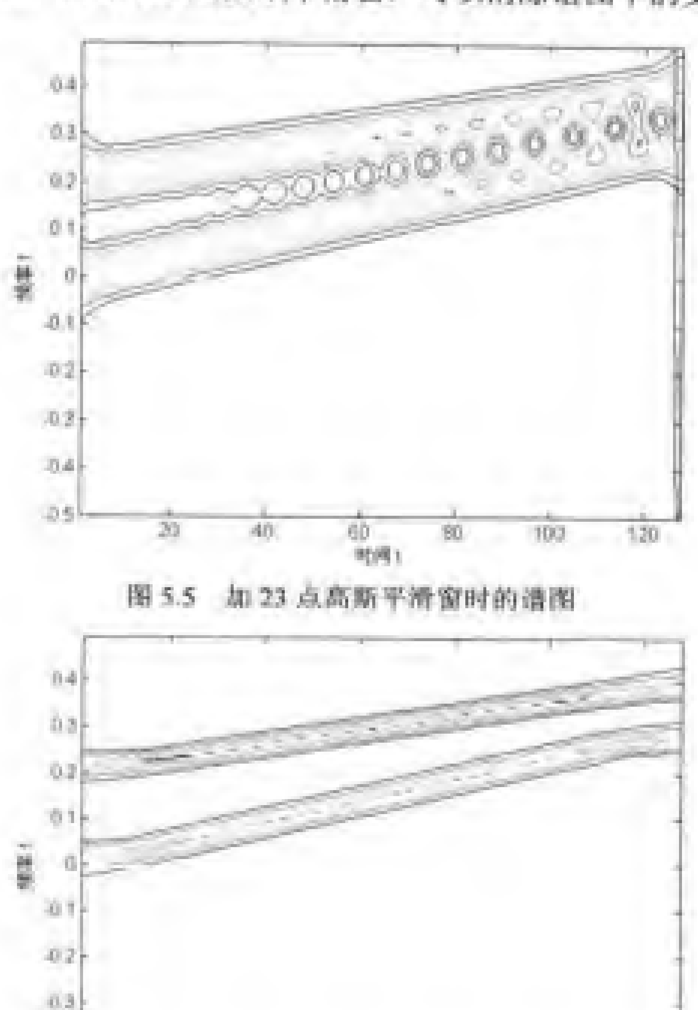


图 5.5　加 23 点高斯平滑窗时的谱图

图 5.6　加 63 点高斯平滑窗时的谱图

5.2　Cohen 类时频分布

由 Cohen 类时频分布的定义可知，核函数对 Wigner-Ville 分布起着一种平滑作用，平滑的目的是抑制 Wigner-Ville 分布的交叉项。从这个意义上说，Cohen 类时频分布是为了减小 Wigner-Ville 分布的交叉项而提出的。由于是平滑，自然在平滑交叉项的同时也会对信号项起抹平作用，从而使时频聚集性降低，因此必须选择核函数的形状和范围，以便在抑制交叉项和时频聚集性两者之间有一个好的兼顾。

5.2.1　Wigner-Ville 分布

典型的 Cohen 类时频分布就是 Wigner-Ville 分布，其在时频分析工具箱中的函数为 tfrwv.m。

tfrwv 函数

【语法格式】

① [tfr,t,f]= tfrwv(x)

② [tfr,t,f]= tfrwv(x,t)

③ [tfr,t,f]= tfrwv(x,t,N)

④ [tfr,t,f]= tfrwv(x,t,N,trace)

【使用说明】计算离散信号 x 的 Wigner-Ville 分布。

输入参数：如果 x 是一个变量，则计算其 Wigner-Ville 分布，如果 x=[x1,x2]，则计算互 Wigner-Ville 分布；t 是时间；N 是频率点数。

输出参数：tfr 返回 Wigner-Ville 分布；t 返回时间坐标；f 返回频率坐标。

例程 5-3：计算线性调频信号的 Wigner-Ville 分布。

f5_3.m

```
%生成线性调频信号
sig=fmlin(256);
%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig);
%显示
figure(1);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
figure(2)
mesh(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

线性调频信号的 Wigner-Ville 分布的计算结果分别如图 5.7 和图 5.8 所示。

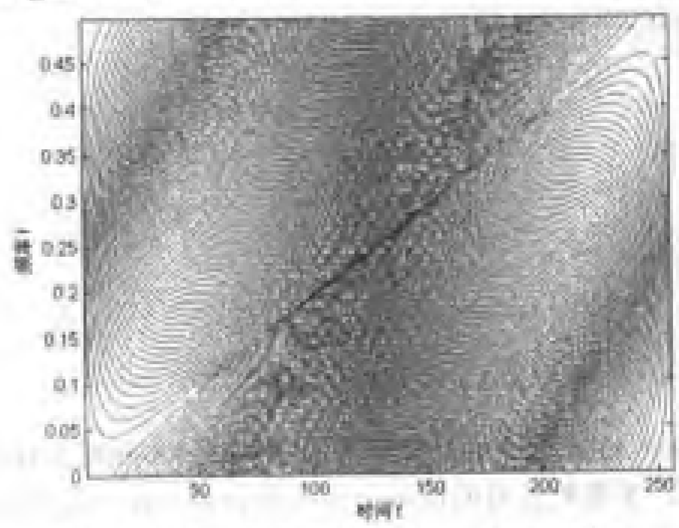


图 5.7 线性调频信号的 WV 分布（等高线图）

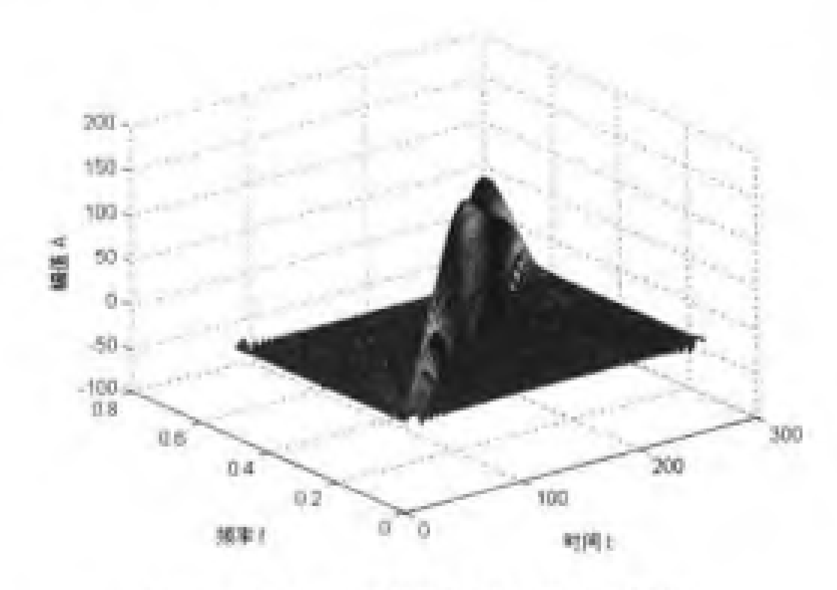


图 5.8　线性调频信号的 WV 分布（三维图）

由图 5.7 可以明显看出，信号的频率是随时间线性变换的，与理论值是一致的，这说明 Wigner-Ville 分布能揭示信号能量在时频面上的分布情况。

例程 5-4：计算多普勒信号的 Wigner-Ville 分布。

f5_4.m

```
%生成多普勒信号
[fm,am,iflaw]=doppler(256,50,13,10,200);
sig=am.*fm;

%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig);

%显示
figure(1);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');

figure(2)
mesh(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

多普勒信号的 Wigner-Ville 分布的计算结果分别如图 5.9 和图 5.10 所示。

由图 5.9 可以看出，多普勒信号的能量分布与理论值存在一定差异，在时频面上本不应有能量的地方存在很多交叉项，这是由于 Wigner-Ville 分布的双线性造成的。

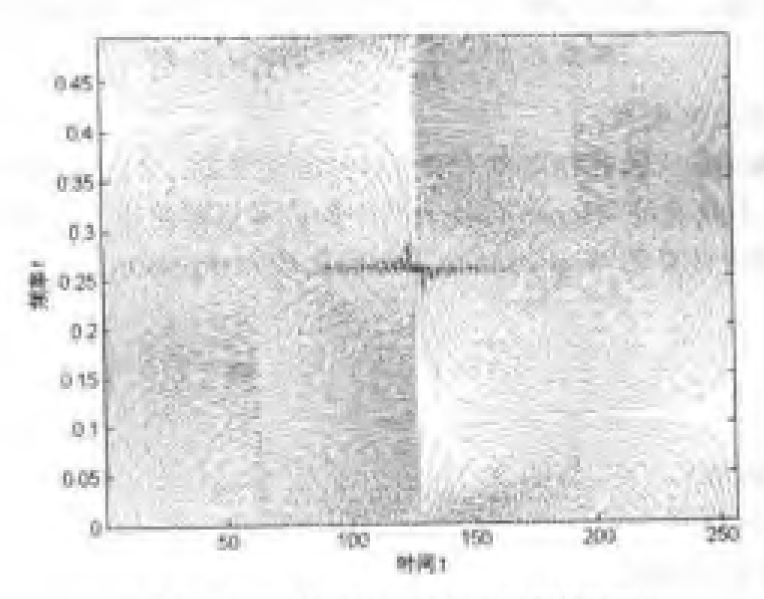


图 5.9 多普勒信号的 WV 分布（等高线图）

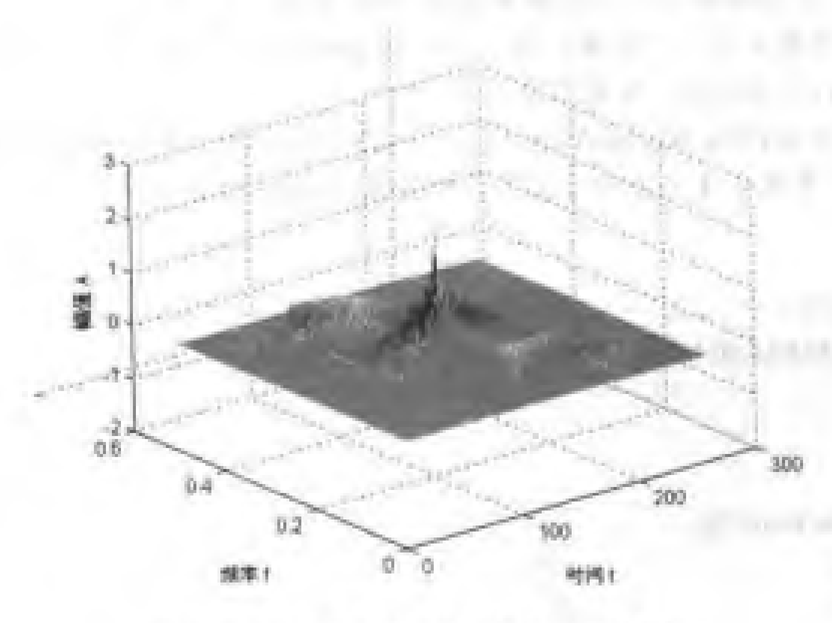


图 5.10 多普勒信号的 WV 分布（三维图）

5.2.2 伪 Wigner–Ville 分布

由于 Wigner-Ville 分布是信号的二次变换，因此它也存在交叉项。但与谱图的交叉项不同，不管分量之间的时频距离大小是多少，交叉项都不会消失。Wigner-Ville 分布交叉项的特点可以归纳为：两个分量会在它们的几何中点处产生第三个交叉项分量，除此之外，交叉项会在连接这两点的直线上产生振荡，振荡频率与这两点之间的距离成正比。

伪 Wigner-Ville 分布定义为

$$PWD_x(t,f)=\int_{-\infty}^{\infty}h(\tau)x(t+\frac{\tau}{2})x^*(t-\frac{\tau}{2})e^{-j2\pi f\tau}d\tau$$

式中 $h(\tau)$是一个矩形窗。上述定义相当于在频率对 Wigner-Ville 分布进行了平滑，即

$$PWD_x(t,f)=\int_{-\infty}^{\infty}H(f-\xi)W_x(t,\xi)d\xi$$

式中 $H(f)$是 $h(\tau)$的傅立叶变换。

由于交叉项是振荡的，因此利用窗平滑可以抑制交叉项。不过这种做法带来的不足就是伪 Wigner-Ville 分布不能满足 Wigner-Ville 分布中的许多重要性质，比如边缘性质等。

时频分析工具箱中计算伪 Wigner-Ville 分布的函数为 tfrpwv.m。

tfrpwv 函数

【语法格式】

① [tfr,t,f]= tfrpwv (x)

② [tfr,t,f]= tfrpwv (x,t)

③ [tfr,t,f]= tfrpwv (x,t,N)

④ [tfr,t,f]= tfrpwv (x,t,N,trace)

【使用说明】计算离散信号 x 的伪 Wigner-Ville 分布。

输入参数：如果 x 是一个变量，则计算其 Wigner-Ville 分布，如果 x=[x1,x2]，则计算互 Wigner-Ville 分布；t 是时间；N 是频率点数。

输出参数：tfr 返回伪 Wigner-Ville 分布；t 返回时间坐标；f 返回归一化的频率坐标。

例程 5-5：计算包含 4 个高斯分量的信号的伪 Wigner-Ville 分布。

f5_5.m

```
%产生非平稳信号
sig=atoms(128,[32,0.15,20,1;96,0.15,20,1;32,0.35,20,1;96,0.35,20,1]);

%时域波形
figure(2)
plot(real(sig),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');

%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig);
%时频表示
figure(3)
contour(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');

%计算伪 Wigner-Ville 分布
sig=hilbert(sig);
[tfr,t,f]=tfrpwv(sig);
```

```
%时频表示
figure(4)
contour(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
```

产生的多分量信号波形如图 5.11 所示，其 Wigner-Ville 分布如图 5.12 所示，而伪 Wigner-Ville 分布如图 5.13 所示。比较图 5.12 和图 5.13 可见：在四分量信号的 Wigner-Ville 分布中存在 6 个交叉项（其中两个发生了重叠），而在伪 Wigner-Ville 分布中则抑制了部分交叉项。

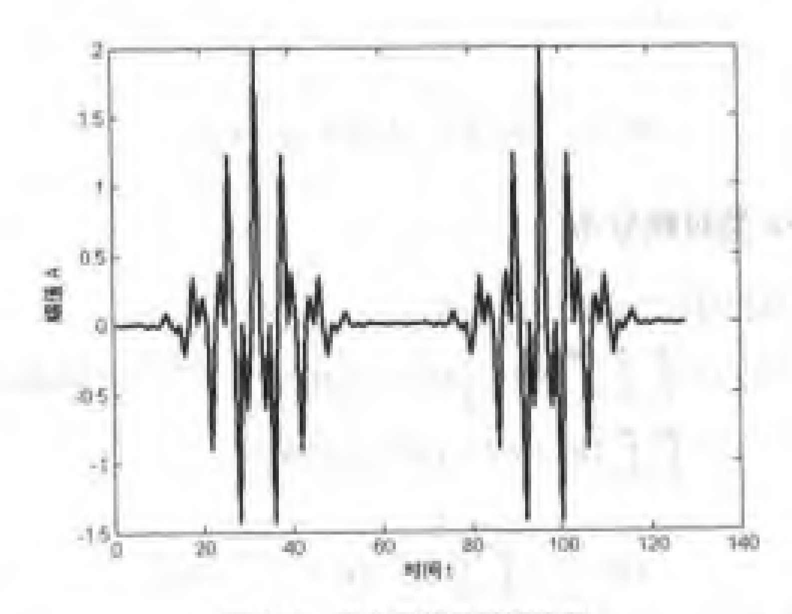


图 5.11　四分量信号时域波形

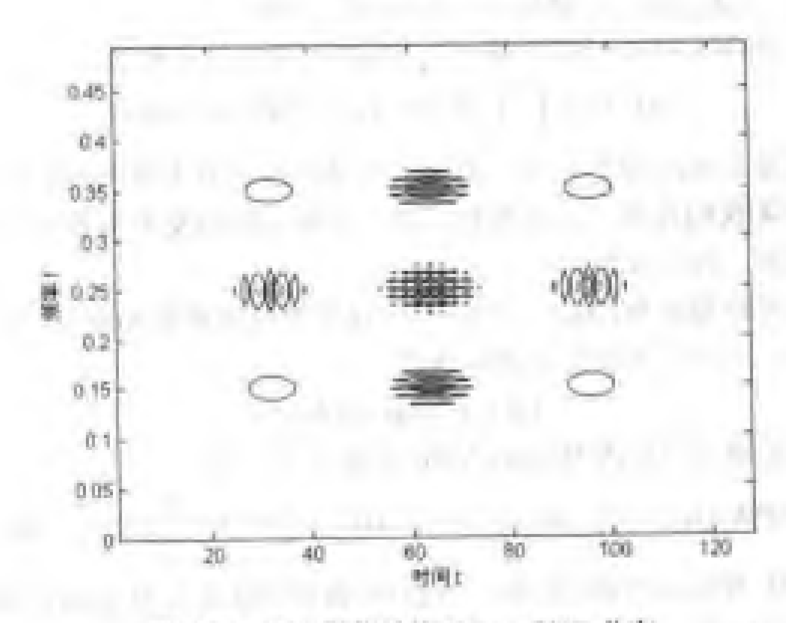


图 5.12　四分量信号的 Wigner-Ville 分布

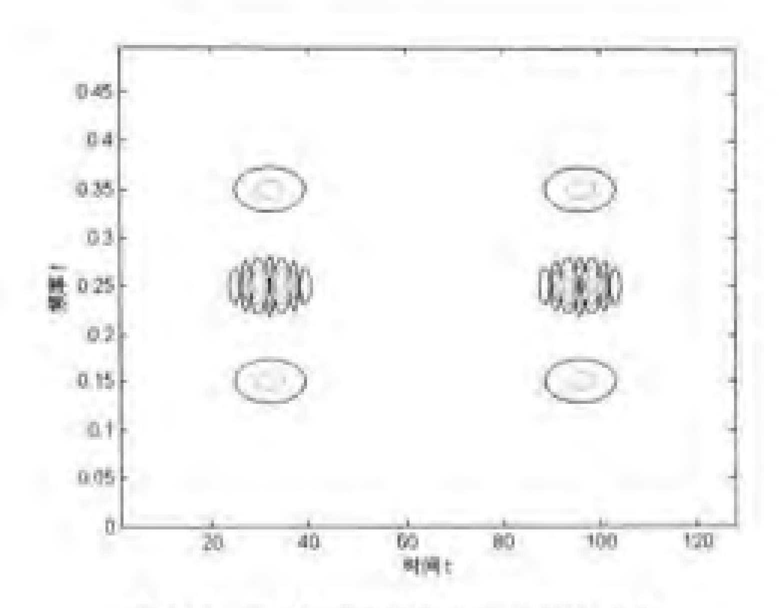


图 5.13 四分量信号的伪 Wigner-Ville 分布

5.2.3 Cohen 类时频分布

Cohen 类时频分布可以写成统一的形式

$$C(t,f)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}z(u+\frac{\tau}{2})z^{*}(u-\frac{\tau}{2})\phi(\tau,v)e^{-j2\pi(tv+\tau f-uv)}dudvd\tau$$

$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\Pi(s-t,\xi-f)W_{z}(s,\xi)dsd\xi$$

其中：

$$\Pi(t,f)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi(\xi,\tau)e^{-j2\pi(f\tau+\xi t)}d\tau d\xi$$

因此Π可以看成是一个平滑函数。同样 Cohen 类时频分布可以理解为平滑后的 Wigner-Ville 分布，当$\phi(\xi,\tau)$=1，就是 Wigner-Ville 分布。

有前面可知，谱图也可以写成平滑后的 Wigner-Ville 分布形式

$$S(t,f)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}W_{h}(s-t,\xi-f)W_{z}(s,\xi)dsd\xi$$

因此，谱图也是 Cohen 类的一种。由这个定义也可以看出谱图的时间分辨率和频率分辨率之间的矛盾：如果我们选择一个短窗 h，则平滑窗会短时宽而宽频宽，导致好的时间分辨率和差的频率分辨率，反之亦然。

上述定义中的平滑窗函数$\Pi(t,f)=W_h(t,f)$仅仅受短时窗函数 $h(t)$的控制。为了在时域和频域独立控制平滑窗，可以如下定义平滑窗函数：

$$\Pi(t,f)=g(t)\cdot H(-f)$$

利用此平滑窗函数可以修改 Wigner-Ville 分布定义，即

$$SPW_{x}(t,f)=\int_{-\infty}^{\infty}h(\tau)\int_{-\infty}^{\infty}g(s-t)x(s+\frac{\tau}{2})x^{*}(s-\frac{\tau}{2})dse^{-j2\pi f\tau}d\tau$$

也称作平滑的伪 Wigner-Ville 分布，当 $g(t)=\delta(t)$时就是伪 Wigner-Ville 分布。时频分析工具箱中计算平滑的伪 Wigner-Ville 分布函数为 tfrspwv.m。

1. Tfrspwv 函数

【语法格式】

① [tfr,t,f]= tfrspwv(x)

② [tfr,t,f]= tfrspwv(x,t)

③ [tfr,t,f]= tfrspwv(x,t,N)

④ [tfr,t,f]= tfrspwv(x,t,N,g)

⑤ [tfr,t,f]= tfrspwv(x,t,N,g,h)

⑥ [tfr,t,f]= tfrspwv(x,t,N,g,h,trace)

【使用说明】计算离散信号 x 的平滑伪 Wigner-Ville 分布。

输入参数：如果 x 是一个变量，则计算其平滑伪 Wigner-Ville 分布，如果 x=[x1,x2]，则计算互平滑伪 Wigner-Ville 分布；t 是时间；N 是频率点数；g 是时域平滑函数；h 是频域平滑函数的时域形式。

输出参数：tfr 返回伪 Wigner-Ville 分布；t 返回时间坐标；f 返回归一化的频率坐标。

例程 5-6：分别计算两分量信号的 Wigner-Ville 分布、伪 Wigner-Ville 分布和平滑伪 Wigner-Ville 分布。

f5_6.m

```
%产生两分量信号
sig=fmconst(128,0.15)+amgauss(128).*fmconst(128,0.4);

%显示
figure(1);
plot(real(sig),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');

%计算 Wigner-Ville 分布
figure(2);
[tfr,t,f]=tfrwv(sig);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');

%计算伪 Wigner-Ville 分布
[tfr,t,f]=tfrpwv(sig);
figure(3)
contour(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');

%计算平滑伪 Wigner-Ville 分布
```

```
[tfr,t,f]=tfrspwv(sig);
%显示
figure(4)
contour(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
```

产生的两分量信号时域波形如图 5.14 所示，其 Wigner-Ville 分布如图 5.15 所示、伪 Wigner-Ville 分布如图 5.16 所示，平滑伪 Wigner-Ville 分布如图 5.17 所示。比较图 5.15～图 5.17 可见：在 Wigner-Ville 分布中能看见两个信号的自主项以及它们之间的交叉项；由于这些交叉项只在时间轴上振荡，因此频域平滑的伪 Wigner-Ville 分布降低了频率分辨率，而且不能抑制交叉项；而平滑伪 Wigner-Ville 分布进行了时域平滑，因此降低了交叉项的影响。

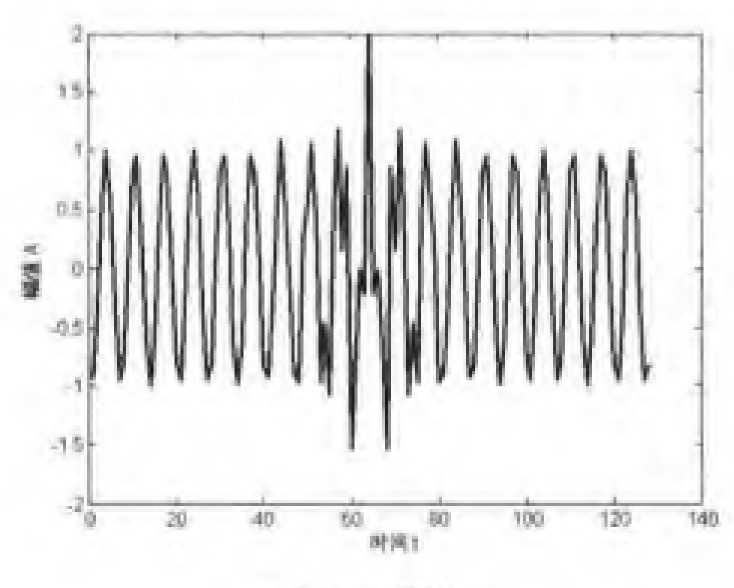


图 5.14 时域波形

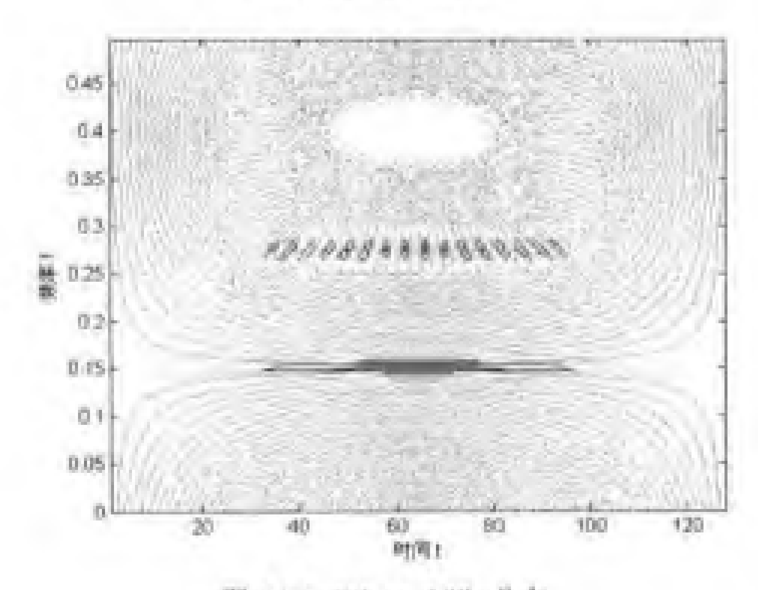


图 5.15 Wigner-Ville 分布

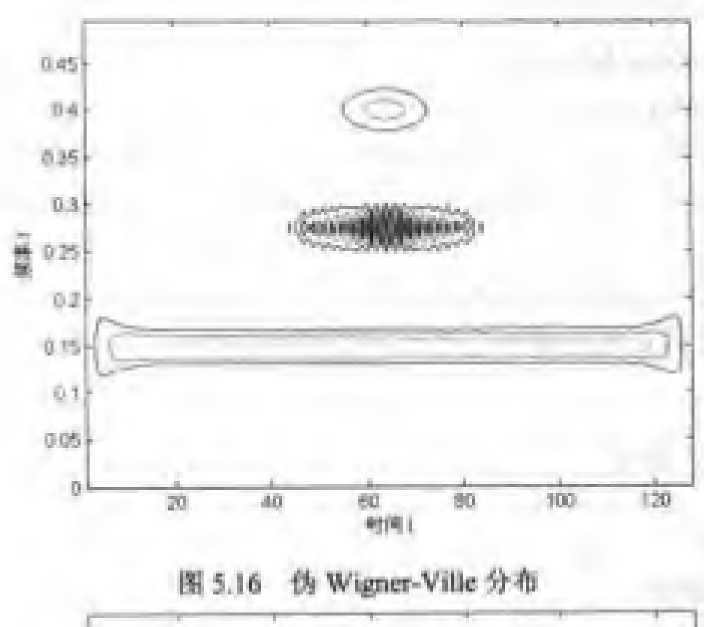


图 5.16 伪 Wigner-Ville 分布

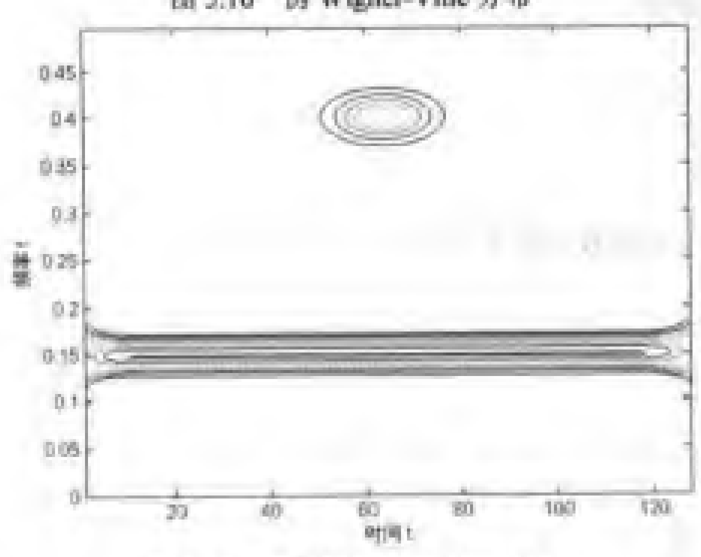


图 5.17 平滑伪 Wigner-Ville 分布

Cohen 类时频分布还可以写成用模糊函数表示的形式，即

$$C_z(t,f)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F(\tau,v)A_z(\tau,v)e^{-j2\pi(tv+f\tau)}dvd\tau$$

式中 F 是Π的二维傅立叶变换。由此定义可以看出，F 相当于一个加权函数，目的是抑制交叉项。

模糊函数定义为

$$A_z(\tau,v)=\int_{-\infty}^{\infty}z(u+\frac{\tau}{2})z^*(u-\frac{\tau}{2})e^{j2\pi vu}dt$$

它可以看成是信号 x 的时频相关性度量。由基本理论我们知道，模糊函数与 Wigner-Ville 分布之间是二维傅立叶变换的关系。时频分析工具箱中计算模糊函数的函数是 ambifunb.m 和 ambifuwb.m。

2. Ambifunb 函数

【语法格式】

① [naf,tau,xi]=ambifunb(x)

② [naf,tau,xi]=ambifunb(x,tau)

③ [naf,tau,xi]=ambifunb(x,tau,N)

④ [naf,tau,xi]=ambifunb(x,tau,N,trace)

【使用说明】计算信号 x 的窄带模糊函数或两个信号之间的互窄带模糊函数。

输入参数：x 是分析信号（如果 $x=[x_1, x_2]$，则计算互窄带模糊函数）；tau 是延迟向量；N 是频率点数。

输出参数：naf 返回多普勒延迟表示；xi 返回多普勒值向量。

例程 5-7：计算二进制相位键移信号的窄带模糊函数。

f5_7.m

```
%产生二进制相位键移信号
sig=anabpsk(256,8);
%计算窄带模糊函数
[naf,tau,xi]=ambifunb(sig);
%显示
contour(2*tau,xi,abs(naf).^2,16);
xlabel('延迟 tau');
ylabel('多普勒值 A');
```

计算得到的窄带模糊函数如图 5.18 所示。

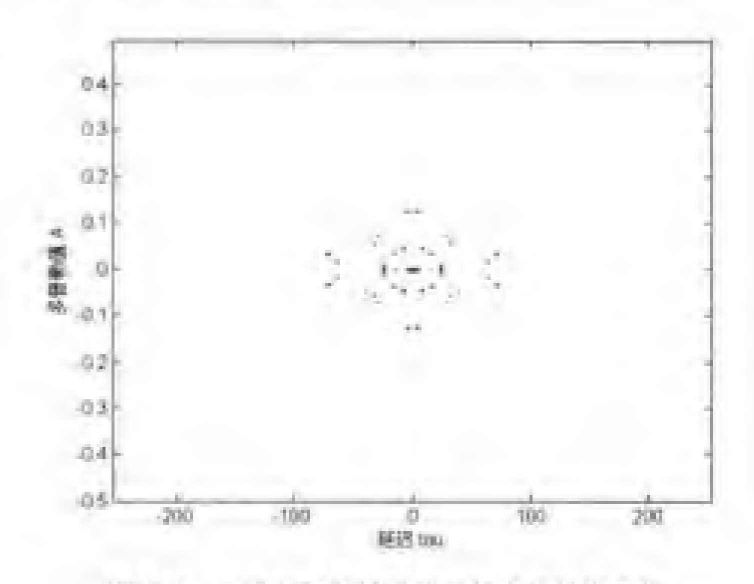


图 5.18　二进制相位键移信号的窄带模糊函数

3. ambifuwb 函数

【语法格式】

① [waf,tau,theta]=ambifuwb(x)

② [waf,tau, theta]=ambifuwb(x,fmin,fmax)

③ [waf,tau, theta]=ambifuwb(x, fmin,fmax,N)

④ [waf,tau, theta]=ambifuwb(x, fmin,fmax,N,trace)

【使用说明】计算信号 x 的宽带模糊函数或两个信号之间的互宽带模糊函数。

输入参数：x 是分析信号（如果 $x=[x_1, x_2]$，则计算互窄带模糊函数）；fmin,fmax 分别对应分析信号频率的上下界；N 是 Mellin 点数。

输出参数：waf 返回模糊函数系数矩阵；tau 返回对应时延的 x 坐标；theta 返回 y 坐标。

例程 5-8：计算二进制相位键移信号的宽带模糊函数。

f5_8.m

```
%产生二进制相位键移信号
sig=anabpsk(256,8);
%计算窄带模糊函数
[waf,tau,theta]=ambifuwb(sig);
%显示
figure(1)
contour(tau,theta,abs(waf.^2));
xlabel('延迟 tau');
ylabel('尺度的对数值 theta');
```

计算得到的宽带模糊函数如图 5.19 所示。

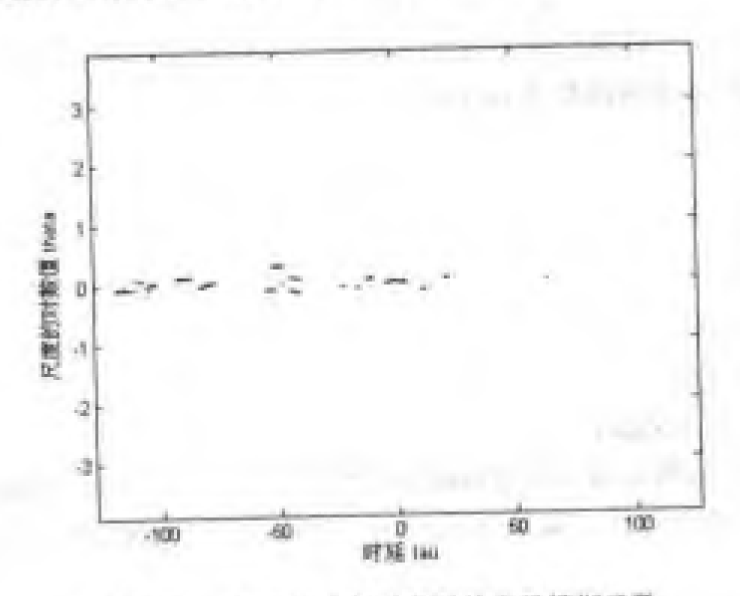


图 5.19　二进制相位键移信号的宽带模糊函数

例程 5-9：比较模糊函数与 Wigner-Ville 分布之间的关系。

f5_9.m

```
%产生信号
N=128;
sig=fmlin(N);
%计算窄带模糊函数
amb=ambifunb(sig);
amb=amb([N/2+1:N 1:N/2],:);
```

```
%模糊函数反变换
ambi=ifft(amb).';
tdr=zeros(N);
tdr(1:N/2,:)=ambi(N/2:N-1,:);
tdr(N:-1:N/2+2,:)=ambi(N/2-1:-1:1,:);
wvd1=real(fft(tdr));
%计算 Wigner-Ville 分布
[wvd2,t,f]=tfrwv(sig);
%比较
diff=max(max(abs(wvd1-wvd2)))
```

计算得到信号模糊函数的反变换与其 Wigner-Ville 分布之间的最大差值为

diff = 2.8422e-014

约等于零，从而验证了理论结果。

5.2.4 其他重要的能量分布

（1）Rihaczek 和 Margenau-Hill 分布

Rihaczek 分布定义为：

$$R_x(t,f)=x(t)x^*(f)e^{-j2\pi ft}$$

它对应于在 Cohen 类时频分布中取 $F(\tau,\nu)=e^{j\pi\nu\tau}$，但是它是复值的能量分布。时频分析工具箱中计算 Rihaczek 分布的函数为 tfrri.m。

1．tfrri 函数

【语法格式】

① [tfr,t,f]=tfrri(x)

② [tfr,t,f]=tfrri(x,t)

③ [tfr,t,f]=tfrri(x,t,N)

④ [tfr,t,f]=tfrri(x,t,N,trace)

【使用说明】计算离散信号 x 的 Rihaczek 分布或两个信号的互 Rihaczek 分布。

输入参数：x 是输入信号，如果 $x=[x_1,x_2]$，则计算互 Rihaczek 分布；N 是频率点数；t 是时间（1:length(x)）。

输出参数：tfr 返回 Rihaczek 分布；t 返回时间坐标；f 返回频率坐标。

例程 5-10：计算线性调频信号的 Rihaczek 分布。

f5_10.m

```
%生成线性调频信号
sig=fmlin(128,0.1,0.4);
%计算 Rihaczek 分布
[tfr,t,f]=tfrri(sig);
%显示
%%等高线图
```

```
figure(1);
contour(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
%%三维图
figure(2)
mesh(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

计算线性调频信号的 Rihaczek 分布等高线如图 5.20 所示，其三维视图如图 5.21 所示。

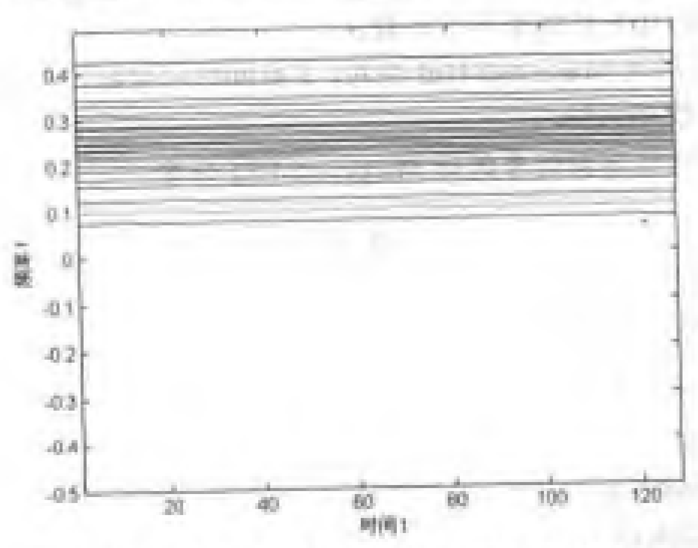


图 5.20 线性调频信号的 Rihaczek 分布（等高线图）

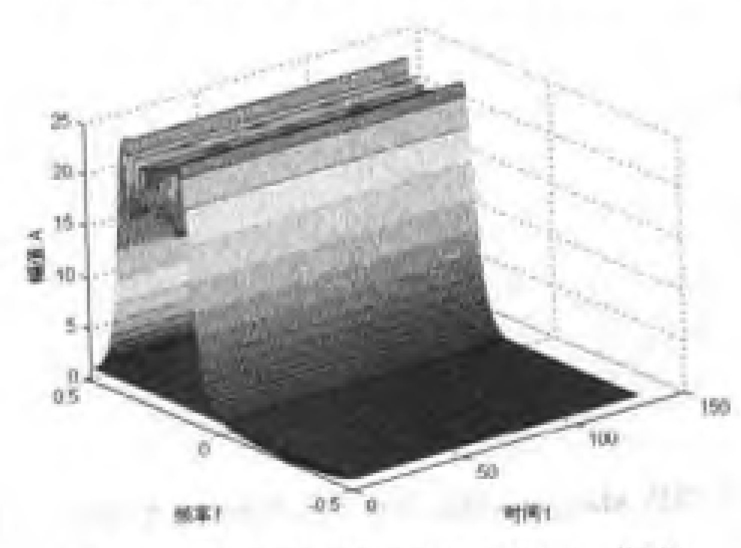


图 5.21 线性调频信号的 Rihaczek 分布（三维图）

Rihaczek 分布的实部也是 Cohen 类的时频分布，称作 Margenau-Hill 分布。时频分析工具箱中计算伪 Margenau-Hill 分布的函数为 tfrpmh.m。

2. tfrpmh 函数

【语法格式】

① [tfr,t,f]= tfrpmh (x)

② [tfr,t,f]= tfrpmh (x,t)

③ [tfr,t,f]= tfrpmh (x,t,N)

④ [tfr,t,f]= tfrpmh (x,t,N,h)

⑤ [tfr,t,f]= tfrpmh (x,t,N,h,trace)

【使用说明】计算离散信号 x 的伪 Margenau-Hill 分布或两个信号的互伪 Margenau-Hill 分布分布。

输入参数：x 是输入信号，如果 $x=[x_1,x_2]$，则计算互伪 Margenau-Hill 分布；N 是频率点数；t 是时间[1:length(x)]; h 是频率平滑函数。

输出参数：tfr 返回伪 Margenau-Hill 分布；t 返回时间坐标；f 返回频率坐标。如果没有输出参数，则调用函数 tfrqview。

例程 5-11：计算线性调频信号的伪 Margenau-Hill 分布。

f5_11.m

```
%生成线性调频信号
sig=fmlin(128,0.1,0.4);
t=1:128;
%窗函数
h=window(63,'Kaiser');
%计算伪 Margenau-Hill 分布
[tfr,t,f]=tfrpmh(sig,t,128,h,1);
%显示
%%等高线图
figure(1);
contour(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
%%三维图
figure(2)
mesh(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

计算线性调频信号的伪 Margenau-Hill 分布等高线如图 5.22 所示，其三维视图如图 5.23 所示。

Rihaczek 和 Margenau-Hill 分布对多分量信号也存在交叉项，但是它们的交叉项与 Wigner-Ville 分布的交叉项不同：位于(t_1,f_1)和(t_2,f_2)处的两个分量信号的交叉项分别位于(t_1,f_2)和(t_2,f_1)，见例程 5-12。

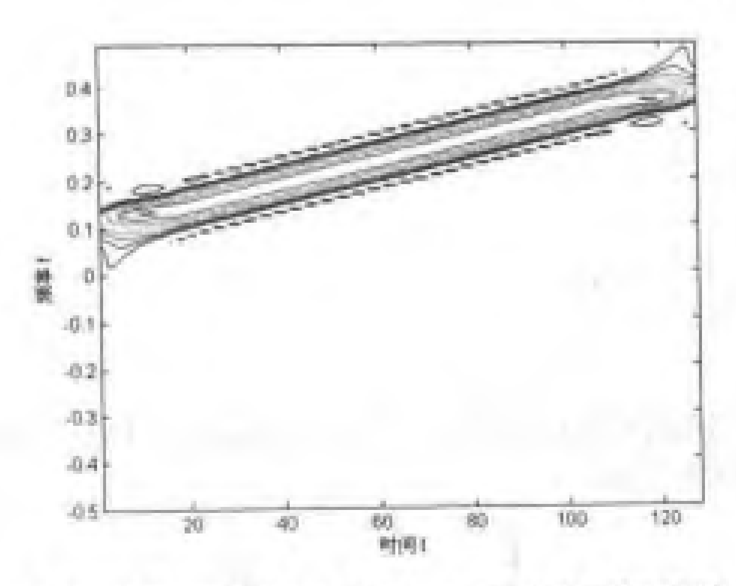


图 5.22 线性调频信号的伪 Margenau-Hill 分布（等高线图）

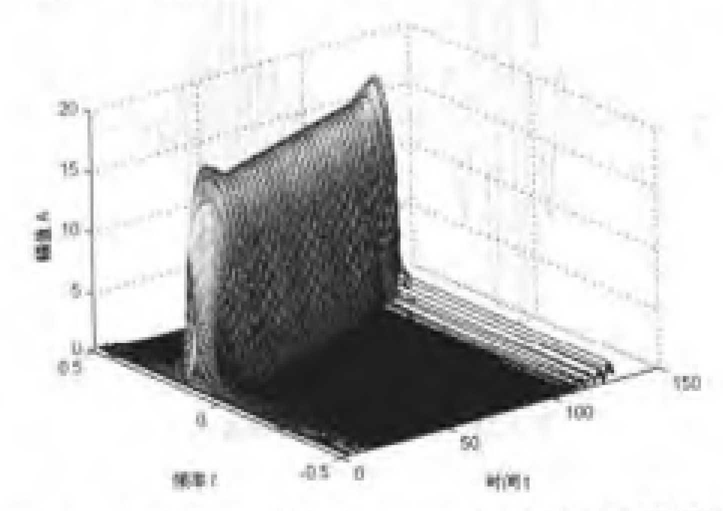


图 5.23 线性调频信号的伪 Margenau-Hill 分布（三维图）

例程 5-12：计算两分量信号的 Margenau-Hill 分布。

f5_12.m

```
%生成两分量信号
sig=atoms(128,[32,0.15,20,1;96,0.32,20,1]);
%时域波形
figure(2)
plot(real(sig),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
%计算 Margenau-Hill 分布
[tfr,t,f]=tfrmh(sig);
%显示
%%等高线图
figure(3);
```

```
contour(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
%%三维图
figure(4)
mesh(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

产生的两分量信号的时域波形如图 5.24 所示，其 Margenau-Hill 分布的等高线图如图 5.25 所示，三维视图如图 5.26 所示。

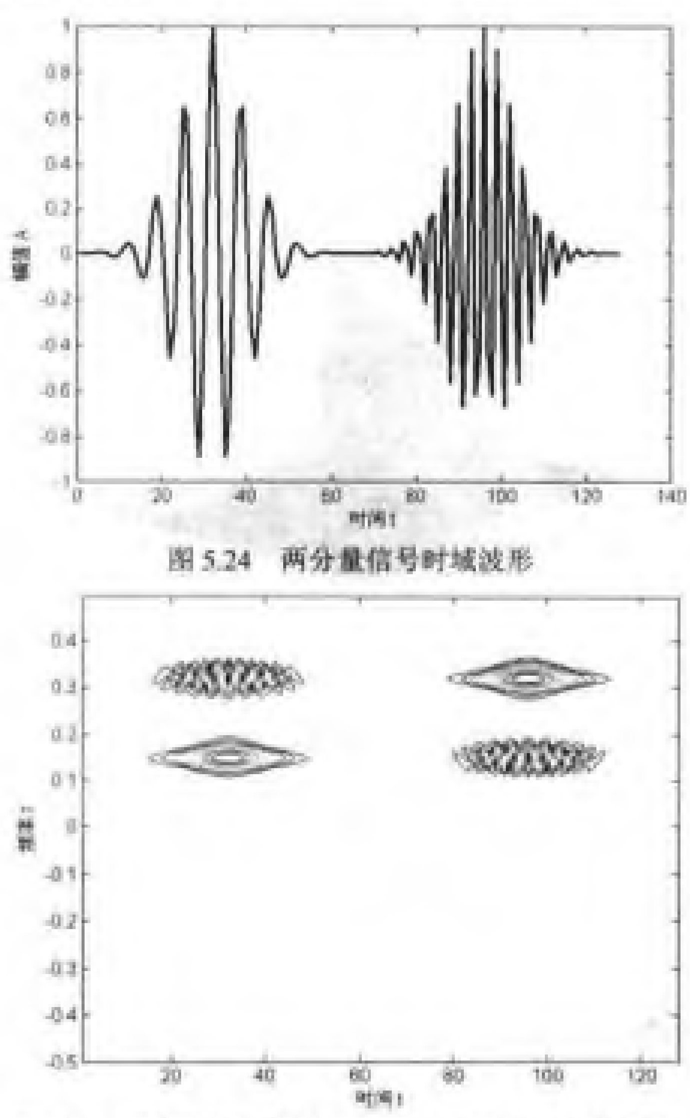


图 5.24　两分量信号时域波形

图 5.25　两分量信号的 Margenau-Hill 分布（等高线图）

（2）Page 分布

Page 分布定义为：

$$P_x(t,f)=\frac{d}{dt}\{|\int_{-\infty} x(u)e^{-j2\pi fu}du|\}$$

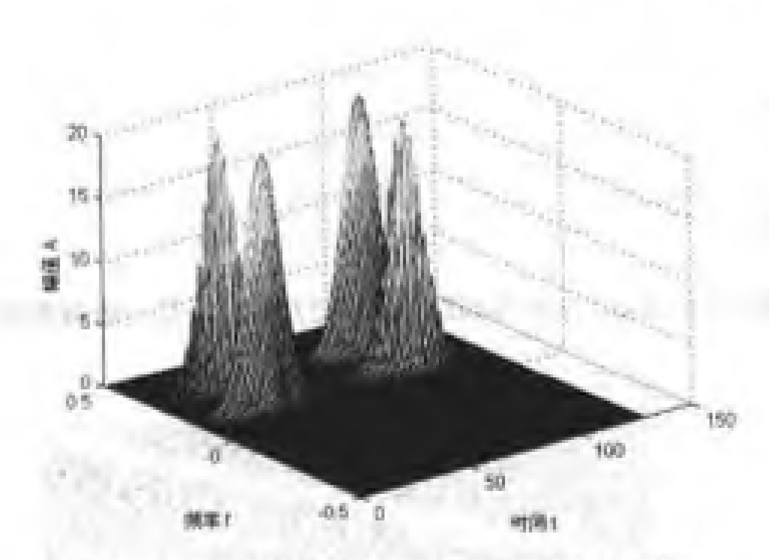


图 5.26 两分量信号的 Margenau-Hill 分布（三维）

它表示信号在时间 t 之前的能量谱密度的微分，也是 Cohen 类时频分布的一种，对应于取 $F(\tau,\nu)=e^{-j\pi\nu|\tau|}$。时频分析工具箱中计算 Page 分布的函数为 tfrpage.m，其加频率平滑窗的函数为 tfrppage.m。

1. tfrpage 函数

【语法格式】

① [tfr,t,f]= tfrpage (x)

② [tfr,t,f]= tfrpage (x,t)

③ [tfr,t,f]= tfrpage (x,t,N)

④ [tfr,t,f]= tfrpage (x,t,N,trace)

【使用说明】计算离散信号 x 的 Page 分布或两个信号的互 Page 分布。

输入参数：x 是输入信号，如果 $x=[x_1,x_2]$，则计算互 Rihaczek 分布；N 是频率点数；t 是时间（1:length(x)）。

输出参数：tfr 返回 Page 分布；t 返回时间坐标；f 返回频率坐标（归一化）。

例程 5-13：计算线性调频信号的 Page 分布。

f5_13.m

```
%生成线性调频信号
sig=fmlin(128,0.1,0.4);
%计算 Page 分布
[tfr,t,f]=tfrpage(sig);
%显示
%%等高线图
figure(1);
contour(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
```

```
%%三维图
figure(2)
mesh(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

计算线性调频信号的 Page 分布等高线如图 5.27 所示，其三维视图如图 5.28 所示。

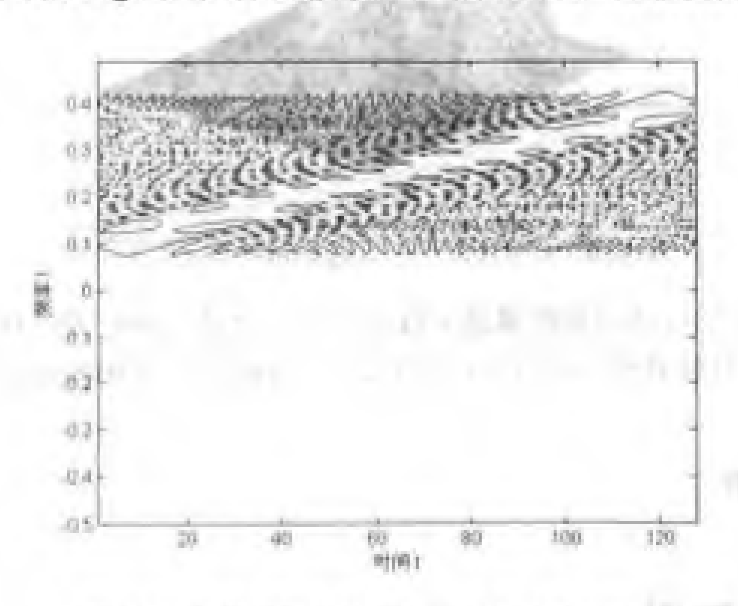


图 5.27 线性调频信号的 Page 分布（等高线）

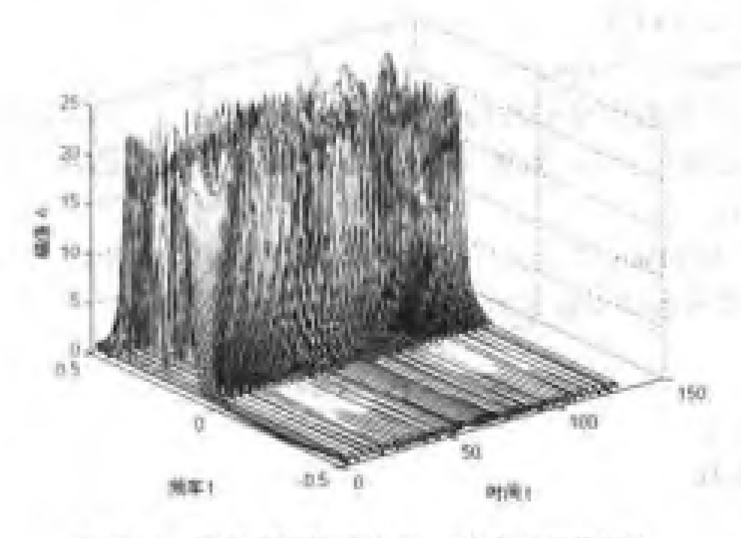


图 5.28 线性调频信号的 Page 分布（三维图）

2. tfrppage 函数

【语法格式】

① [tfr,t,f]= tfrppage (x)

② [tfr,t,f]= tfrppage (x,t)

③ [tfr,t,f]= tfrppage (x,t,N)

④ [tfr,t,f]= tfrppage (x,t,N,h)

⑤ [tfr,t,f]= tfrppage (x,t,N,h,trace)

【使用说明】计算离散信号 x 的伪 Page 分布或两个信号的互伪 Page 分布。

输入参数：x 是输入信号，如果 x=[x_1,x_2]，则计算互 Rihaczek 分布；N 是频率点数；t 是时间（1:length(x)）；h 是频率平滑函数。

输出参数：tfr 返回 Rihaczek 分布；t 返回时间坐标；f 返回频率坐标。

例程 5-14：计算线性调频信号的伪 Page 分布。

f5_14.m

```
%生成线性调频信号
sig=fmlin(128,0.1,0.4);
%计算伪 Page 分布
[tfr,t,f]=tfrppage(sig);
%显示
%%等高线图
figure(1);
contour(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
%%三维图
figure(2)
mesh(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

计算线性调频信号的伪 Page 分布等高线如图 5.29 所示，其三维视图如图 5.30 所示。

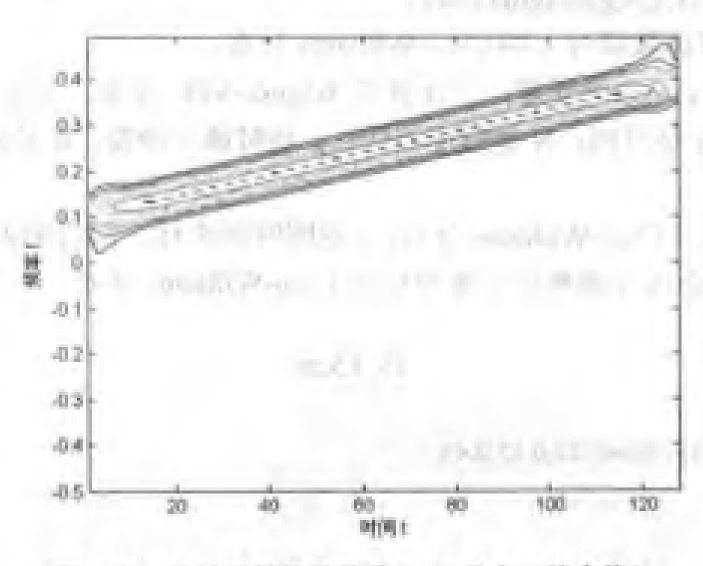


图 5.29　线性调频信号的伪 Page 分布（等高线）

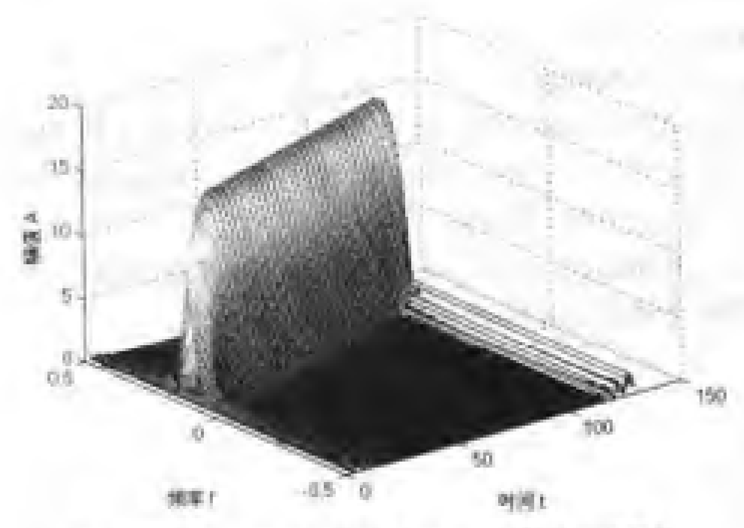


图 5.30 线性调频信号的伪 Page 分布（三维图）

（3）Choi-Williams 分布

在 Cohen 类时频分布统一表达式中取核函数为指数函数，则得到 Choi-Williams 分布的定义，它也是 Wigner-Ville 分布的加窗平滑定义，具有更好地抑制交叉项的效果。时频分析工具箱中计算 Choi-Williams 分布的函数为 tfrcw.m。

tfrcw 函数

【语法格式】

① [tfr,t,f]= tfrcw (x)

② [tfr,t,f]= tfrcw (x,t)

③ [tfr,t,f]= tfrcw (x,t,N,g)

④ [tfr,t,f]= tfrcw (x,t,N,g,h)

⑤ [tfr,t,f]= tfrcw (x,t,N,g,h,sigma)

⑥ [tfr,t,f]= tfrcw (x,t,N,g,h,sigma,trace)

【使用说明】计算离散信号 x 的 Choi-Williams 分布。

输入参数：如果 x 是一个变量，则计算其 Wigner-Ville 分布，如果 x=[x1,x2]，则计算互 Choi-Williams 分布；t 是时间；N 是频率点数；g 是时域平滑窗；h 是频域平滑窗；sigma 核函数的宽度。

输出参数：tfr 返回 Choi-Williams 分布；t 返回时间坐标；f 返回频率坐标。

例程 5-15：计算两线性调频信号叠加后的 Choi-Williams 分布。

f5_15.m

```
%生成信号
sig=fmlin(128,0.05,0.3)+fmlin(128,0.15,0.4);

%时域波形
figure(1)
```

```
plot(real(sig),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');

%时域窗函数
g=window(9,'Kaiser');
%频域窗函数
h=window(27,'Kaiser');
t=1:128;

%计算 Choi-Williams 分布
[tfr,t,f]=tfrcw(sig,t,128,g,h,3.6,1);

%显示
%%等高线图
figure(2);
contour(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');

%%三维图
figure(3)
mesh(t,f,abs(tfr));
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

产生信号的时域波形如图 5.31 所示，其 Choi-Williams 分布的等高线如图 5.32 所示，三维视图如图 5.33 所示。

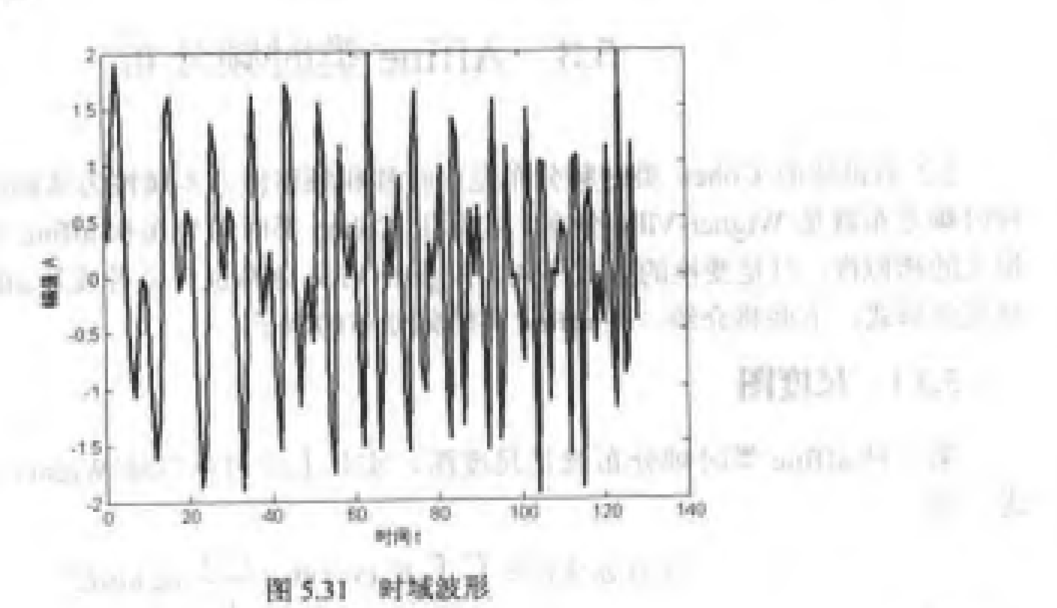


图 5.31 时域波形

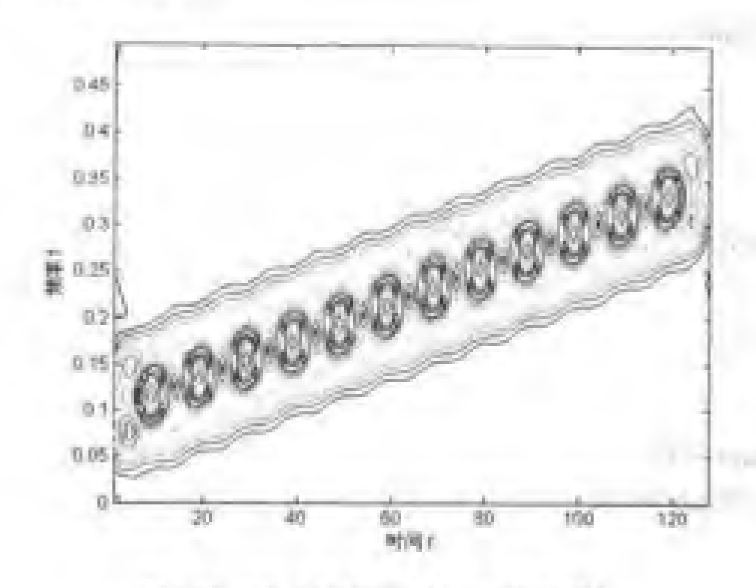


图 5.32　Choi-Williams 分布（等高线）

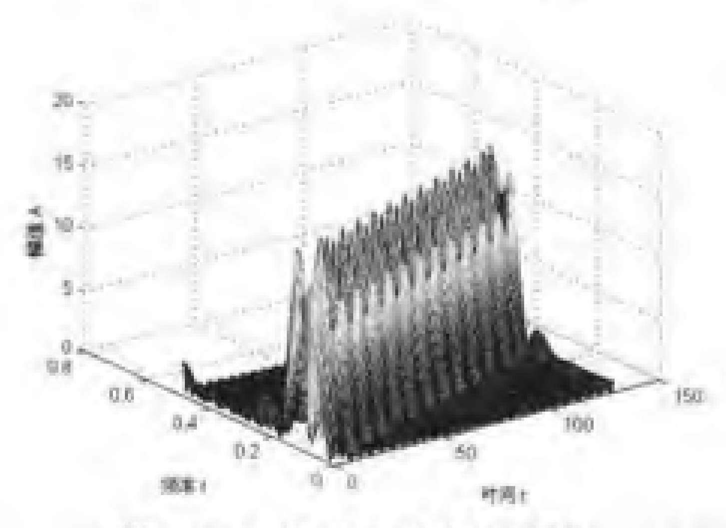


图 5.33　Choi-Williams 分布（三维图）

5.3　Affine 类时频分布

5.2 节讲述的 Cohen 类时频分布是以时移和频移协方差属性为基础的，这类中重要的一种时频分布就是 Wigner-Ville 分布。实际上 Cohen 类时频分布和 affine 类时频分布之间存在很大的相似性，只是变换的方式不同。Wigner-Ville 分布也可以看成是 affine 类时频分布的一种具体形式，下面将介绍一些 affine 类时频分布的例子。

5.3.1　尺度图

第一种 affine 类时频分布就是尺度图，实际上它可以写成 Wigner-Ville 分布的平滑形式，即

$$|T_x(t,a;\Psi)|^2=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}W_x(s,\xi)W_\Psi\left(\frac{s-t}{a},a\xi\right)dsd\xi$$

因此尺度图对应于 affine 类时频分布中的$\Pi(t,f)=W_\Psi(t,f)$。通过比较谱图的定义，我们可以发现尺度图是谱图在 affine 类中的对应者。

同谱图相似，利用小波可以定义尺度图

$$SC_z(a,b)=|WT_z(a,b)|^2=\frac{1}{|a|}|\int_{-\infty}^{\infty}z(t)h^*(\frac{t-b}{a})dt|^2$$

时频分析工具箱中计算尺度图的函数是 tfrscalo.m。

tfrscalo 函数

【语法格式】

① [tfr,t,f,wt]= tfrscalo(x)

② [tfr,t,f,wt]= tfrscalo(x,t)

③ [tfr,t,f,wt]= tfrscalo(x,t,wave)

④ [tfr,t,f,wt]= tfrscalo(x, t,wave,fmin,fmax)

⑤ [tfr,t,f,wt]= tfrscalo(x, t,wave,fmin,fmax,N)

⑥ [tfr,t,f,wt]= tfrscalo(x, t,wave,fmin,fmax,N)

【使用说明】计算信号 x 的尺度图。

输入参数：x 是要分析的信号，其解析形式为 z=hilbert(real(*x*))；t 是时间，为 1:length(*x*)；wave 是 Morlet 分析小波的半长度；fmin，fmax 分别对应分析信号频率的上下界，必须大于 0 而小于 0.5。

输出参数：tfr 返回时频矩阵；t 返回时间坐标；f 返回归一化的频率坐标；wt 返回包含相应小波变换的复矩阵。

例程 5-16：计算两种信号的尺度图，第一个信号是冲击信号，第二个信号是由两个正弦信号叠加而成。

f5_16.m

```
%产生非平稳信号
%%信号 1:冲击信号
sig1=unapulse(128);
%%信号 2：两个正弦信号的叠加
sig2=fmconst(128,0.15)+fmconst(128,0.35);

%计算信号 1 的尺度图
[tfr,t,f,wt]=tfrscalo(sig1,1:128,6,0.05,0.45,128,1);
%时频分布
figure(1)
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');

%计算信号 2 的尺度图
[tfr,t,f,wt]=tfrscalo(sig2,1:128,6,0.05,0.45,128,1);
```

```
%时频分布
figure(2)
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
```

冲击信号的尺度图如图 5.34 所示，第二个信号的尺度图如图 5.35 所示。

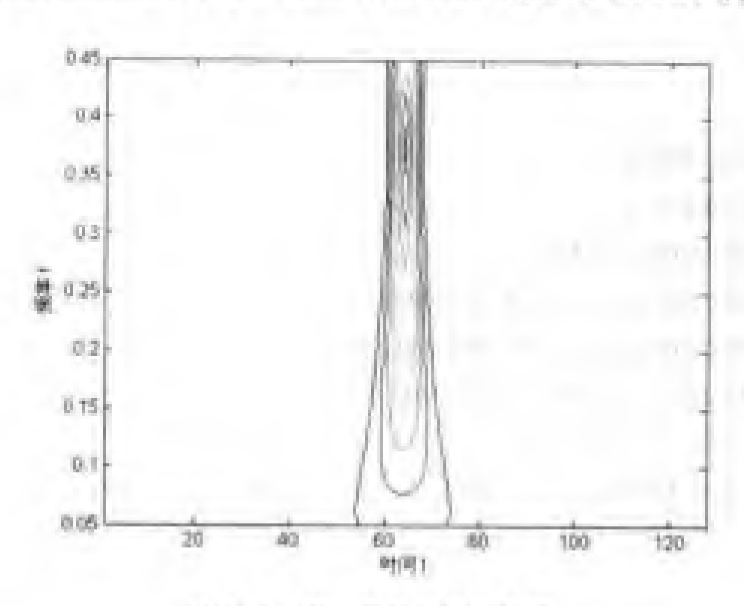


图 5.34　第一个信号的尺度图

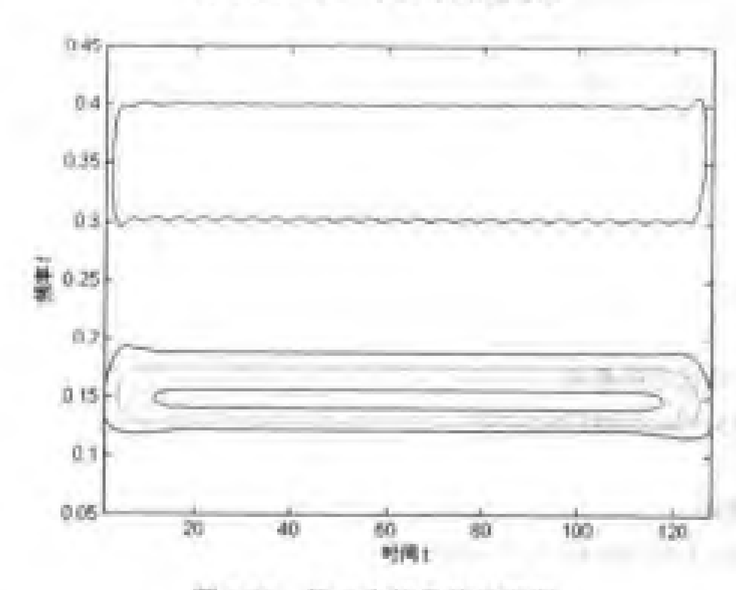


图 5.35　第二个信号的尺度图

5.3.2　乘积核分布

一般的“尺度等于频率的倒数”约定可以进行扩展，得到其他的不同于 Wigner-Ville 分布的时频分布。假设考虑如下形式的核函数

$$\Phi(\xi,\tau)=\Phi(\xi\tau)e^{-j2\pi f_0\tau}$$

式中 f_0 是某个非零频率，那么可以得到 Cohen 类和 Affine 类时频分布之间的等式关系

$$\Omega_x(t,a;\Phi)=C_x(t,\frac{f_0}{a};\Phi)$$

相应的表示，尤其是 Wigner-Ville 分布等是这两类的共同元素。

5.3.3 Affine 类平滑伪 Wigner 分布

为了克服尺度图时域和频域分辨率之间的相互制约性，一种方法就是使用一个时域和频域分离的平滑函数，由此得到的分布称作 Affine 类平滑伪 Wigner 分布，即

$$ASPW_x(t,a)=\frac{1}{a}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h(\frac{\tau}{a})g(\frac{s-t}{a})x(s+\frac{\tau}{2})x^*(s-\frac{\tau}{2})dsd\tau$$

上述定义使得可以通过选择窗 g 和 h，独立地确定时间和尺度分辨率。

Wigner-Ville 分布具有最好的时间和频率分辨率，但是也带来严重的交叉项干扰；而尺度图具有最差的时间和频率分辨率，但几乎没有交叉项，因此 Affine 类平滑伪 Wigner 分布是这两者之间的一种很好的综合。

时频分析工具箱计算 Affine 类平滑伪 Wigner 分布的函数为 tfrspaw.m

tfrspaw 函数

【语法格式】

① [tfr,t,f]=tfrspaw(x)

② [tfr,t,f]=tfrspaw(x,t)

③ [tfr,t,f]=tfrspaw(x,t,k)

④ [tfr,t,f]=tfrspaw(x,t,k,nh0)

⑤ [tfr,t,f]=tfrspaw(x,t,k,nh0,ng0)

⑥ [tfr,t,f]=tfrspaw(x,t,k,nh0,ng0,fmin,fmax)

⑦ [tfr,t,f]=tfrspaw(x,t,k,nh0,ng0,fmin,fmax,N)

⑧ [tfr,t,f]=tfrspaw(x,t,k,nh0,ng0,fmin,fmax,N,trace)

【使用说明】计算 affine 类平滑伪 Wigner 分布。

输入参数：x 是要分析的解析信号；t 是时间常数（1：length(x)）；k 是分布的标记，k=-1 表示平滑伪 Unterberger 分布，k=0 表示平滑伪 Bertrand 分布，k=1/2 表示平滑伪 D-Flandrin 分布，k=2 表示 affine 平滑伪 Wigner-Ville 分布；nh0 是分析小波的半长度；ng0 是时域平滑窗的半长度；fmin，fmax 分别对应分析信号频率的上下界。

输出参数：tfr 返回时频矩阵；f 返回归一化的频率；如果没有输出参数，将调用函数 tfrqview.m

例程 5-17：计算 altes 信号的 affine 类平滑伪 Wigner 分布。

f5_17.m

```
%生成阿 altes 信号
sig=altes(64,0.1,0.5);
%计算 affine 类平滑伪 Wigne 分布
[tfr,t,f]=tfrspaw(sig);
%显示
```

```
figure(1);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');

figure(2)
mesh(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

计算结果分别如图 5.36 和图 5.37 所示。

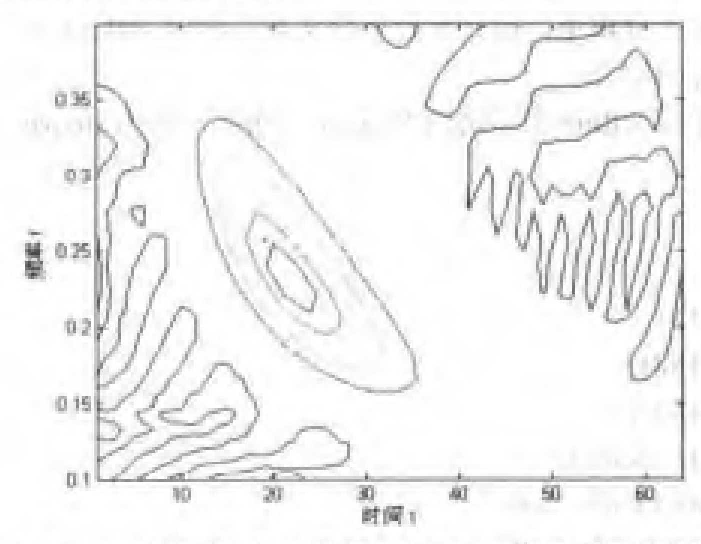


图 5.36 altes 信号的 affine 类平滑伪 Wigner 分布（等高线图）

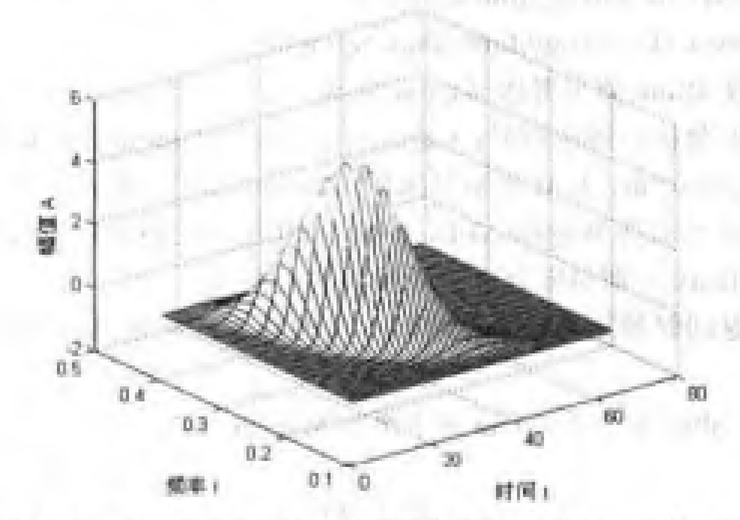


图 5.37 altes 信号的 affine 类平滑伪 Wigner 分布（三维图）

例程 5-18：比较四分量信号的不同时频分布结果。

f5_18.m

```
N=128;
```

```
%生成四分量信号
t1=N/4;
t2=3*N/4;
t3=t1;
t4=t2;
f1=0.15;
f2=f1;
f3=0.35;
f4=f3;
t=1:N;
T=sqrt(2*N);
sig1=amgauss(N,t1,T).*fmconst(N,f1,t1);
sig2=amgauss(N,t2,T).*fmconst(N,f2,t2);
sig3=amgauss(N,t3,T).*fmconst(N,f3,t3);
sig4=amgauss(N,t4,T).*fmconst(N,f4,t4);
sig=sig1+sig2+sig3+sig4;
%时域波形
figure(1)
plot(1:N,real(sig));
xlabel('时间 t');
ylabel('实部幅值 A');
%频率上下限
fmin=0.05;
fmax=0.45;

%计算尺度图
[tfr,t,f]=tfrscalo(sig,t,16,fmin,fmax,N);
Max=max(max(tfr));
V=[0.1 0.3 0.5 0.7 0.9]*Max;
figure(2)
contour(t,f,tfr,V);
xlabel('时间 t');
ylabel('频率 f');
%计算 affine 类平滑伪 Wigner 分布
%%设置参数
Np=8;
Ng0=round(fliplr(linspace(2,N/10,Np/2-1)));
Nh0=round(linspace(N/4,3*N/4,Np/2-1));
[tfr,t,f]=tfrspaw(sig,t,0,Nh0(1),Ng0(1),fmin,fmax,N,1);
Max=max(max(tfr));
V=[0.1 0.3 0.5 0.7 0.9]*Max;
figure(3)
contour(t,f,tfr,V);
```

```
xlabel('时间 t');
ylabel('频率 f');
%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig);
Max=max(max(tfr));
V=[0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9]*Max;
figure(4)
contour(t,f,tfr,V);
axis([1 N fmin fmax]);
xlabel('时间 t');
ylabel('频率 f');
```

产生的四高斯分量信号的时域波形如图 5.38 所示，它的尺度图如图 5.39 所示，affine 类平滑伪 Wigner 分布如图 5.40 所示，一般的 Wigner-Ville 分布如图 5.41 所示。

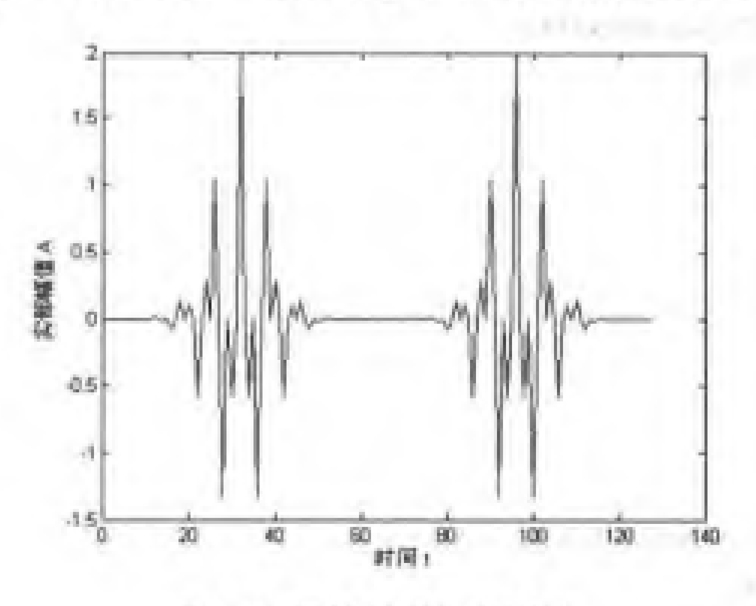


图 5.38　四高斯分量的时域波形

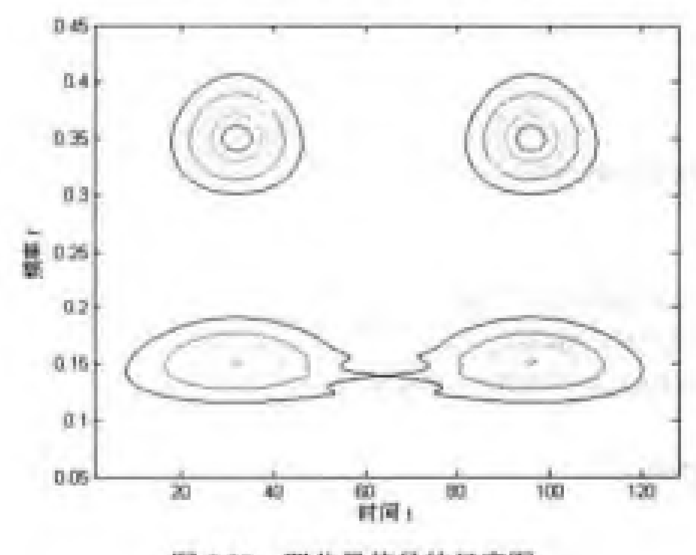


图 5.39　四分量信号的尺度图

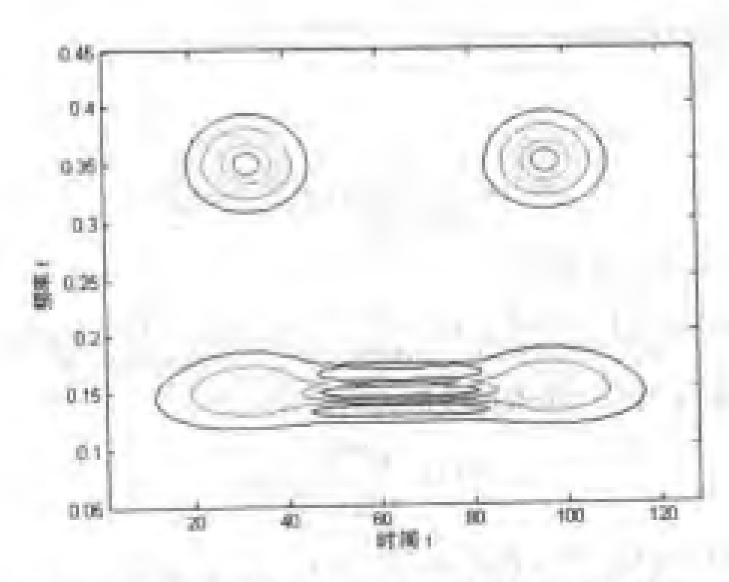


图 5.40 四分量信号的 affine 类平滑伪 Wigner 分布

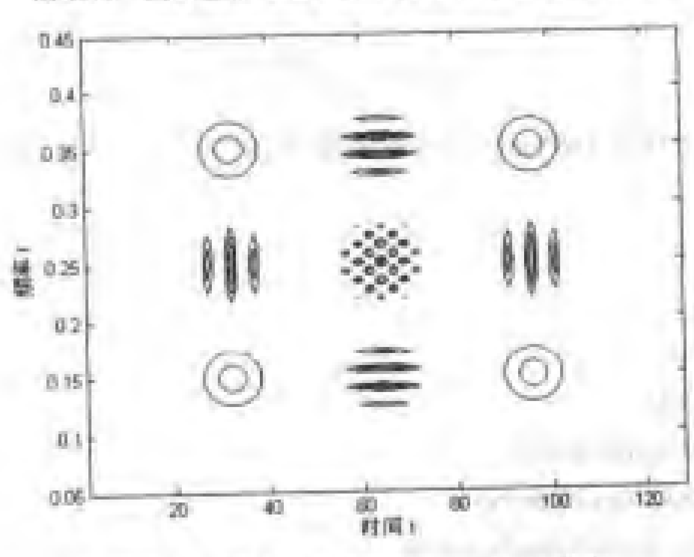


图 5.41 四分量信号的 Wigner-Ville 分布

比较图 5.39～图 5.41 可见：对于四分量信号，Wigner-Ville 分布具有最好的分辨率，但存在严重的交叉项；尺度图具有最差的分辨率，但没有交叉项；affine 类平滑伪 Wigner 分布是上两种时频分布优点的综合。

5.3.4 局部化双频核函数分布

在 affine 类时频分布中有一种很有用的子类，它们的核函数在双频面内完全位于某个曲线 $f=H(v)$ 上，即

$$\Psi(nu, f)=G(v)\delta(f-H(v))-\Phi(v,\tau)=G(v)e^{j2\pi H(v)\tau}$$

其中 $G(v)$ 是任意函数。相应的时间-尺度分布，也称作局部化双频核函数分布，定义为

$$\Omega_x(t,a;\Pi)=\frac{1}{|a|}\int_{-\infty}^{\infty}G(v)X\left(\frac{H(v)-v/2}{a}\right)X^*\left(\frac{H(v)+v/2}{a}\right)e^{-j2\pi vt/a}dv$$

和乘积核函数分布一样，通过替换 $a=f_0/f$，可以将每一个时间-尺度分布对应成一个时频分布，即 $C_x(t,f;\Phi)=\Omega_x(t,f_0/f;\Phi)$

下面将给出局部化双频核函数分布的一些特殊例子。

（1）Bertrand 分布

假设我们定义

$$G(v)=\frac{v/2}{\sinh(v/2)},H(v)=\frac{v}{2}\coth(\frac{v}{2})$$

则得到 Bertrand 分布，定义为

$$B_x(t,a)=\frac{1}{|a|}\int_{-\infty}^{\infty}\frac{v/2}{\sinh(v/2)}X[\frac{ve^{-v/2}}{2a\sinh(v/2)}]\times X^*[\frac{ve^{v/2}}{2a\sinh(v/2)}]e^{-j2\pi t v}dv$$

Bertrand 分布的群延迟是一个双曲线形式的，且

$$X(v)=\frac{e^{j\Phi_x(v)}}{\sqrt{v}}U(v)$$

式中 $\Phi_x(v)=-2\pi[vt_0+\alpha\log\frac{v}{v_r}]\Rightarrow B_x(t,a=\frac{v_0}{v})=v\delta(t-t_x(v))U(v)$

群延迟定义为

$$t_x(v)=-\frac{1}{2\pi}\frac{\mathrm{d}\Phi_x(v)}{\mathrm{d}v}$$

时频分析工具箱中计算 Bertrand 分布的函数为 tfrbert.m。

tfrbert 函数

【语法格式】

① [tfr,t,f]=tfrbert(x)

② [tfr,t,f]=tfrbert(x,t)

③ [tfr,t,f]=tfrbert(x,t,fmin,fmax)

④ [tfr,t,f]=tfrbert(x,t,fmin,fmax,N)

⑤ [tfr,t,f]=tfrbert(x,t,fmin,fmax,N,trace)

【使用说明】计算 Bertrand 分布。

输入参数：x 是分析的解析信号；t 是时间常数；fmin，fmax 分别是解析信号频率的上下限。

输出参数：tfr 返回时频矩阵；f 返回归一化的频率；如果没有输出参数，将调用函数 tfrqview。

例程 5-19：计算具有双曲线群延迟信号的 Bertrand 分布。

f5_19.m

```
%生成信号
N=128;
sig=gdpower(N);
figure(1);
plot(1:N,real(sig));
xlabel('时间 t');
ylabel('幅值 A');
```

```
%计算 bertrand 分布
[tfr,t,f]=tfrbert(sig,1:128,0.01,0.22);
%显示
figure(2);
contour(t,f,tfr);
xlabel('时间  t');
ylabel('频率  f');
figure(3)
mesh(t,f,tfr);
xlabel('时间  t');
ylabel('频率  f');
zlabel('幅值  A');
```

产生的具有双曲线群延迟信号的时域波形如图 5.42 所示，其 Bertrand 分布分别如图 5.43、图 5.44 所示。

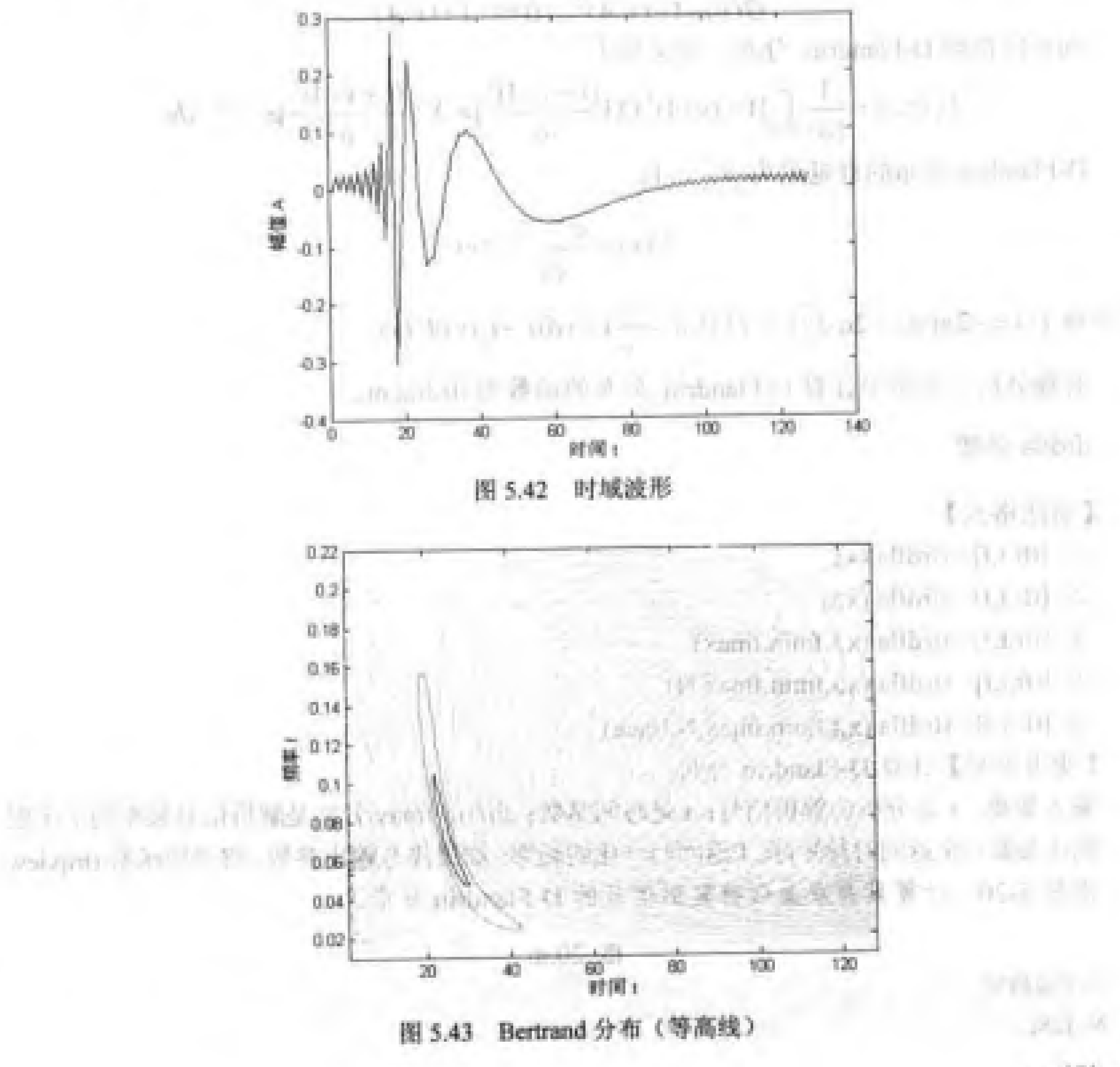


图 5.42　时域波形

图 5.43　Bertrand 分布（等高线）

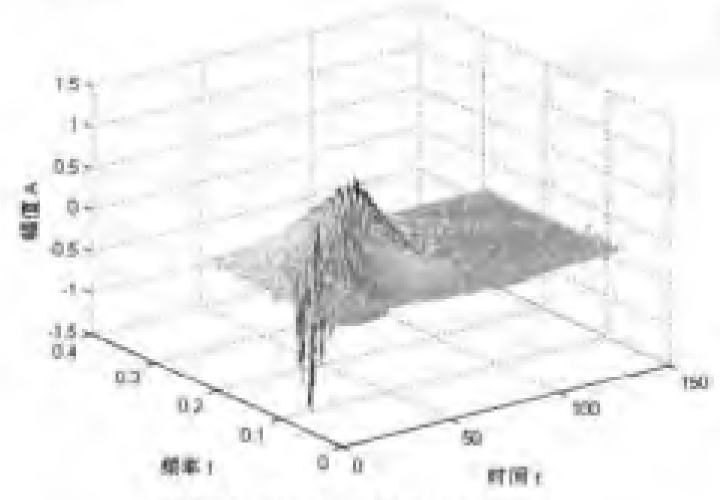


图 5.44　Bertrand 分布（三维图）

（2）D-Flandrin 分布

假设我们定义

$$G(v)=1-(v/4)^2,\ H(v)=1+(v/4)^2$$

则可以得到 D-Flandrin 分布，定义如下：

$$D_x(t,a)=\frac{1}{|a|}\int_{-\infty}^{\infty}[1-(v/4)^2]X[\frac{(1-v/4)^2}{a}]\times X^*[\frac{(1+v/4)^2}{a}]e^{-j2\pi vt/a}dv$$

D-Flandrin 分布的群延迟为 $1/\sqrt{v}$，且

$$X(v)=\frac{e^{j\Phi_x(v)}}{\sqrt{v}}U(v)$$

式中 $\Phi_x(v)=-2\pi[vt_0+2\alpha\sqrt{v}]\Rightarrow D_x(t,a=\frac{v_0}{v})=v\delta(t-t_x(v))U(v)$

时频分析工具箱中计算 D-Flandrin 分布的函数为 tfrdfla.m。

tfrdfla 函数

【语法格式】

① [tfr,t,f]= tfrdfla (x)

② [tfr,t,f]= tfrdfla (x,t)

③ [tfr,t,f]= tfrdfla (x,t,fmin,fmax)

④ [tfr,t,f]= tfrdfla (x,t,fmin,fmax,N)

⑤ [tfr,t,f]= tfrdfla (x,t,fmin,fmax,N,trace)

【使用说明】计算 D-Flandrin 分布。

输入参数：x 是分析的解析信号；t 是时间常数；fmin，fmax 分别是解析信号频率的上下限。

输出参数：tfr 返回时频矩阵；f 返回归一化的频率；如果没有输出参数，将调用函数 tfrqview。

例程 5-20：计算具有双曲线群延迟信号的 D-Flandrin 分布。

f5_20.m

```
%生成信号
N=128;
```

```
sig=gdpower(N,1/2);
figure(1);
plot(1:N,real(sig));
xlabel('时间 t');
ylabel('幅值 A');
%计算 D-Flandrin 分布
[tfr,t,f]=tfrdfla(sig,1:N,0.01,0.22,N,1);

%显示
figure(2);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
figure(3)
mesh(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

产生的具有双曲线群延迟信号的时域波形如图 5.45 所示，其 D-Flandrin 分布分别如图 5.46、图 5.47 所示。

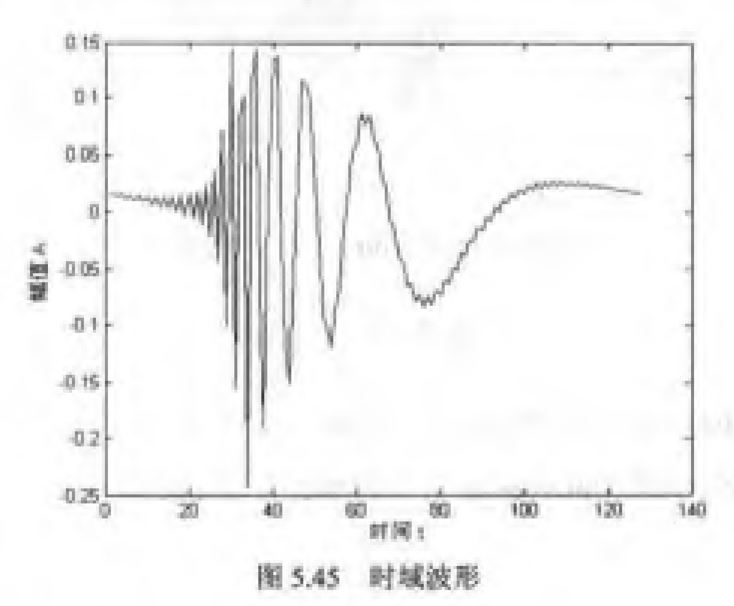


图 5.45 时域波形

（3）Unterberger 分布

假设我们定义

$$G(v)=1, H(v)=\sqrt{1+(v/2)^2}$$

则得到 Unterberger 分布，定义为

$$U_x^{(a)}(t,a)=\frac{1}{|a|}\int_0^{\infty}(1+\frac{1}{\alpha^2})X(\frac{\alpha}{a})X^*(\frac{1}{\alpha a})e^{j2\pi(\alpha-1/\alpha)\frac{t}{a}}d\alpha$$

Unterberger 分布的群延迟为 $1/v^2$，且

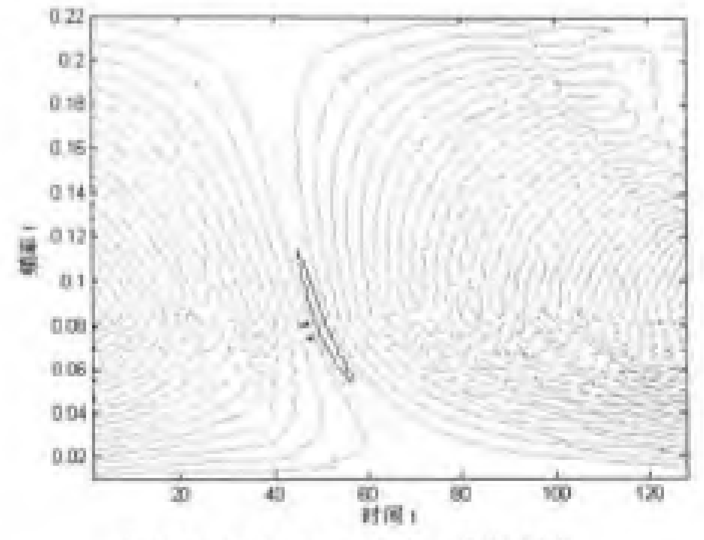


图 5.46　D-Flandrin 分布（等高线图）

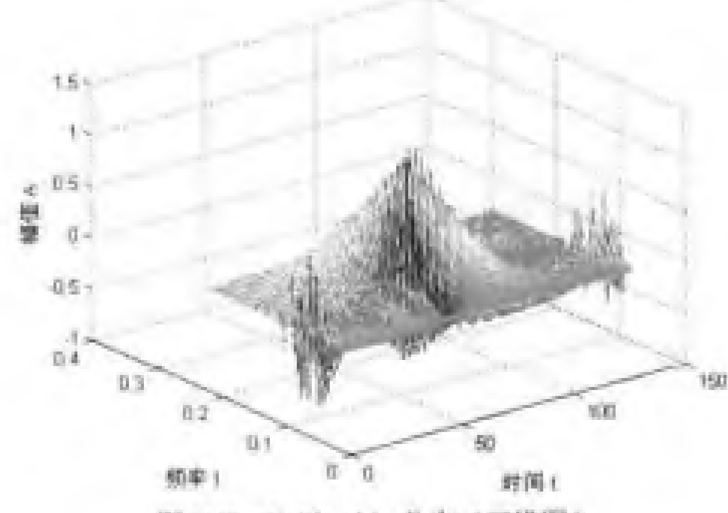


图 5.47　D-Flandrin 分布（三维图）

$$X(v) = \frac{e^{j\Phi_v(v)}}{\sqrt{v}} U(v)$$

式中 $\Phi_v(v) = -2\pi[vt_0 - \alpha / v] \Rightarrow U_v^{(n)}(t, a = \frac{v_0}{v}) = v\delta(t - t_v(v))U(v)$。

时频分析工具箱中计算 Unterberger 分布的函数为 tfrunter.m。

tfrunter 函数

【语法格式】

① [tfr,t,f]= tfrunter (x)

② [tfr,t,f]= tfrunter (x,t)

③ [tfr,t,f]= tfrunter (x,t,form)

④ [tfr,t,f]= tfrunter (x,t,form,fmin,fmax)

⑤ [tfr,t,f]= tfrunter (x,t,form,fmin,fmax,N)

⑥ [tfr,t,f]= tfrunter (x,t,form,fmin,fmax,N,trace)

【使用说明】计算 Unterberger 分布。

输入参数：x 是分析的解析信号；t 是时间常数；form= 'A'表示主动分布，form='P'表示被动分布；fmin，fmax 分别是解析信号频率的上下限。

输出参数：tfr 返回时频矩阵；f 返回归一化的频率；如果没有输出参数，将调用函数 tfrqview。

例程 5-21：计算具有双曲线群延迟信号的 Unterberger 分布。

f5_21.m

```
%生成信号
N=128;
sig=gdpower(N,-1);
figure(1);
plot(1:N,real(sig));
xlabel('时间 t');
ylabel('幅值 A');
%计算 D-Flandrin 分布
[tfr,t,f]=tfrunter(sig,1:N,'A',0.01,0.22,172,1);

%显示
figure(2);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
figure(3)
mesh(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

产生的具有双曲线群延迟信号的时域波形如图 5.48 所示，其 Unterberger 分布分别如图 5.49、图 5.50 所示。

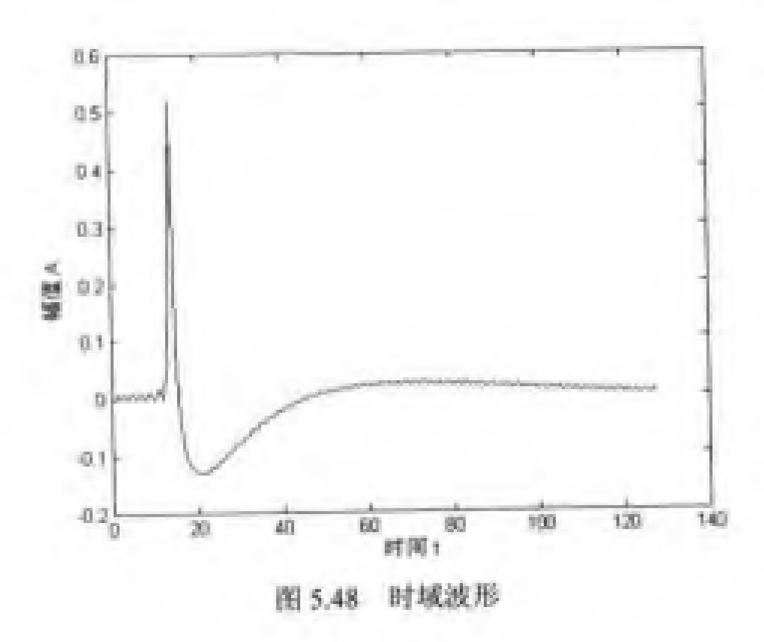


图 5.48　时域波形

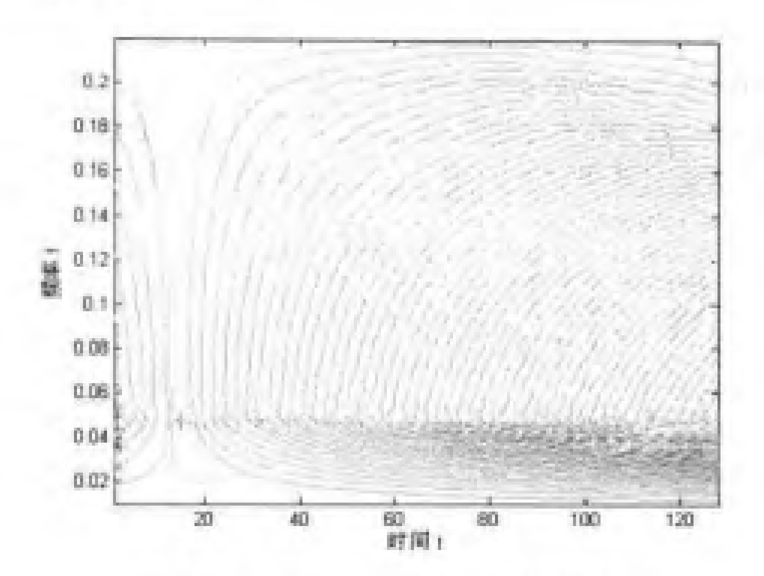


图 5.49　Unterberger 分布（等高线图）

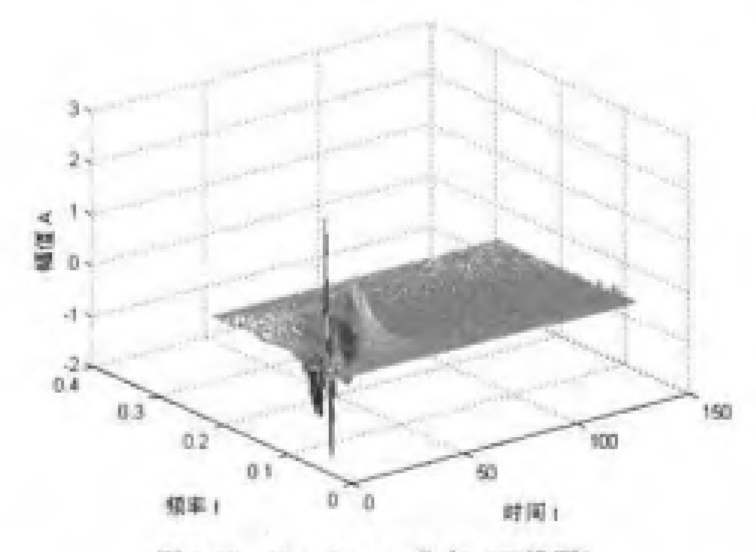


图 5.50　Unterberger 分布（三维图）

第6章 时频分布处理

除了核函数之外，是否还有别的方法能够改善时频分布的性能呢？答案是肯定的，作为平滑手段的一种补充，对时频分布作某些处理可以改善时频分布的时频聚集性，同时减小交叉项，其中一种有效的方法就是通过对信号进行重排，以提高信号分量的时频聚集性，称作重排方法，它适用于一大类的时频分布。

本章主要内容包括：

- 时频分布的重排；
- 时频图像的信息提取。

6.1 时频分布的重排

第5章介绍的是双线性时频分布，它们提供了很多不同的方法来分析非平稳信号。但是在应用这些方法的过程中，存在一个关键的问题，即可解释性，这意味着双线性时频分布应该有一个好的时域和频域聚集性，没有干扰项。但是双线性时频分布存在固有干扰项的缺点，为此人们研究了多种方法来抑制交叉项，其中一类特别的方法就是重排方法。

6.1.1 谱图的重排

重排方法的最初目的是为了改善谱图的效果。实际上和其他双线性能量分布一样，谱图也不可避免地面临着需要在降低交叉项和提高信号分量局部效果之间有一个好的兼顾。

根据前面的定义，谱图看作是信号的 Wigner-Ville 分布和分析窗的 Wigner-Ville 分布之间的二维卷积，即

$$S_x(t,f)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}W_x(s,\xi)W_h(t-s,f-\xi)dsd\xi \qquad (6.1.1)$$

式中 h 是窗函数，W_h 是 h 的 Wigner-ville 分布。

因此，上述定义的分布降低了信号 Wigner-Ville 分布的交叉项，不过它是以时间和频率分辨率为代价的。但是仔细观察上式可以发现，$W_h(t-s, f-\xi)$ 在点 (t, f) 附近构成了一个时频域，在这个区域中对信号的 Wigner-Ville 分布值进行了加权平均。重排原理的关键在于这些值不必以 (t,f) 作为时频域的几何中心而对称分布。因此加权平均不应该位于点 (t,f)，而应该是时频域的重心，这样更能表示信号的局部能量。

重排方法的过程如下：它将谱图在任何点 (t, f) 处计算的值转换到另一点 $(\hat{t},\hat{f})$，这一

点是信号围绕点（t, f）的能量分布的重心，即

$$\hat{t} = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} sW_h(t-s, f-\xi)W_x(s,\xi)dsd\xi}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W_h(t-s, f-\xi)W_x(s,\xi)dsd\xi} \tag{6.1.2}$$

$$\hat{f} = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \xi W_h(t-s, f-\xi)W_x(s,\xi)dsd\xi}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W_h(t-s, f-\xi)W_x(s,\xi)dsd\xi} \tag{6.1.3}$$

这样得到重排的谱图，它在任何点(t', f')处的值等于重排到这一点的所有谱图值的和，即

$$S_x^{(r)}(t', f') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S_x(t,f)\delta(t'-\hat{t})\delta(f'-\hat{f})dtdf \tag{6.1.4}$$

这种新分布的一个最有趣的性质是它还使用了短时傅立叶变换的相位信息，这可以从下面的重排算子看得出来：

$$\hat{t} = -\frac{d\Phi_x(t,f)}{df} \tag{6.1.5}$$

$$\hat{f} = f + \frac{d\Phi_x(t,f)}{dt} \tag{6.1.6}$$

式中$\Phi_x(t,f)$是 x 短时傅立叶变换的相位，即$\Phi_x(t,f)=\arg(\mathrm{STFT}_x(t,f))$。不过这些表达式不利于实际应用，可以用下式替代：

$$\hat{t} = t - \mathrm{Re}\{\frac{\mathrm{STFT}_x(t,f;T_h)\mathrm{STFT}_x^*(t,f;h)}{|\mathrm{STFT}_x(t,f;h)|^2}\} \tag{6.1.7}$$

$$\hat{f} = f - \mathrm{Im}\{\frac{\mathrm{STFT}_x(t,f;D_h)\mathrm{STFT}_x^*(t,f;h)}{|\mathrm{STFT}_x(t,f;h)|^2}\} \tag{6.1.8}$$

式中 $T_h(t)=t\times h(t)$以及 $D_h(t) = \frac{dh}{dt}(t)$ 是平滑窗函数。

最后需要强调的是，虽然重排的谱图不再是双线性的，但它仍然满足时移和频移不变性和非负性质。而且由于 Wigner-Ville 分布对线性调频和冲击信号具有最优的时频聚集性，因此任何重排的谱图也满足以下性质：

$$\begin{aligned} x(t) = Ae^{j(f_0t+\alpha t^2/2)} &\Rightarrow \hat{f} = f_0 + \alpha t \\ x(t) = A\delta(t-t_0) &\Rightarrow \hat{t} = t_0 \end{aligned} \tag{6.1.9}$$

时频分析工具箱中计算重排谱图的函数为 tfrrsp.m。

tfrrsp 函数

【语法格式】

① [tfr,rtfr,hat]= tfrrsp (x)

② [tfr,rtfr,hat]= tfrrsp (x,t)

③ [tfr,rtfr,hat]= tfrrsp (x,t,N)

④ [tfr,rtfr,hat]= tfrrsp (x,t,N,h)

⑤ [tfr,rtfr,hat]= tfrrsp (x,t,N,h,trace)

【使用说明】计算谱图和它重排后的结果。

输入参数：x 是分析的信号；N 是频率点数；h 是频率平滑窗。

输出参数：tfr 返回谱图值；rtfr 返回重排谱图值；hat 返回重排向量的复数矩阵。当没有输出参数时调用函数 tfrqview.m。

例程 6-1：计算线性调频信号的重排谱图。

f6_1.m

```
%产生线性调频信号
sig=fmlin(128,0.1,0.4);
%设置窗函数
h=window(17,'Kaiser');
%获取坐标
[tfr,t,f]=tfrsp(sig);
%计算重排谱图
[tfr,rtfr,hat]=tfrrsp(sig,1:128,64,h,1);

%显示谱图
figure(1);
contour(t,2*f(1:length(f)/2),tfr);
xlabel('时间 t');
ylabel('频率 f');
%%三维图

figure(2)
%显示重排谱图
contour(t,2*f(1:length(f)/2),rtfr);
xlabel('时间 t');
ylabel('频率 f');
```

计算得到线性调频信号的谱图如图 6.1 所示，其重排谱图如图 6.2 所示，比较可见，重排后的谱图具有更好的时频聚集性。

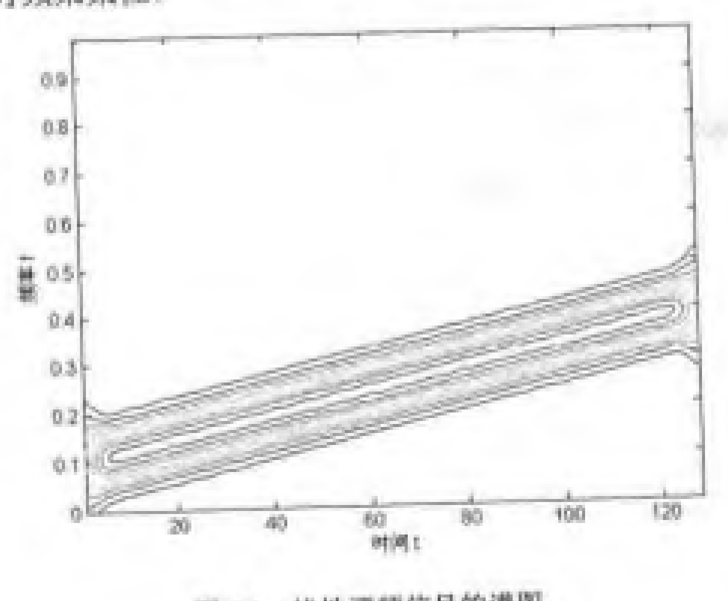


图 6.1 线性调频信号的谱图

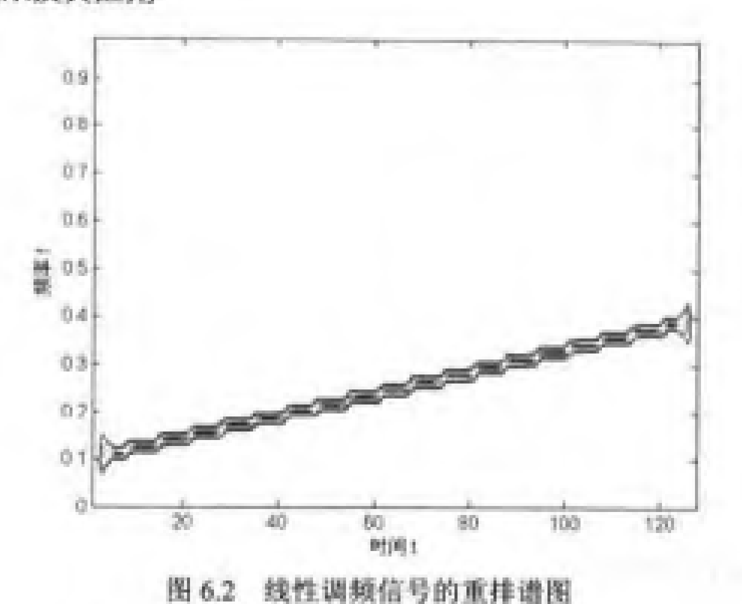


图 6.2　线性调频信号的重排谱图

例程 6-2：计算两分量信号的重排谱图。

f6_2.m

```
%产生两分量信号
N=128;
%%信号分量 1
[sig1,ifl1]=fmsin(N,0.15,0.45,100,1,0.4,-1);
%%信号分量 2
[sig2,ifl2]=fmhyp(N,[1,0.5],[32,0.05]);
sig=sig1+sig2;
%时域波形
plot(real(sig),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
%获取坐标
[tfr,t,f]=tfrsp(sig);
%计算重排谱图
[tfr,rtfr,hat]=tfrrsp(sig);
%显示谱图
figure(2);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
%%三维图
figure(3)
%显示重排谱图
contour(t,f,rtfr);
xlabel('时间 t');
ylabel('频率 f');
```

产生的两分量信号的时域波形如图6.3所示，其谱图如图6.4所示，重排谱图如图6.5所示，比较可见，重排后的谱图具有更好的时频聚集性。

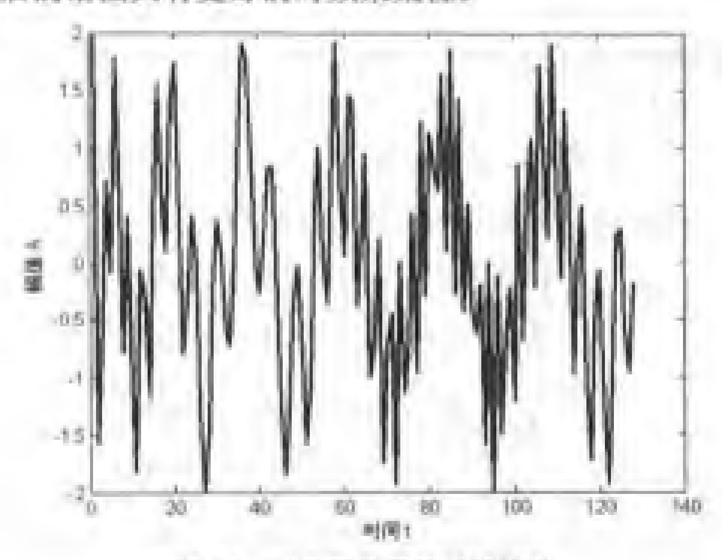


图6.3 两分量信号的时域波形

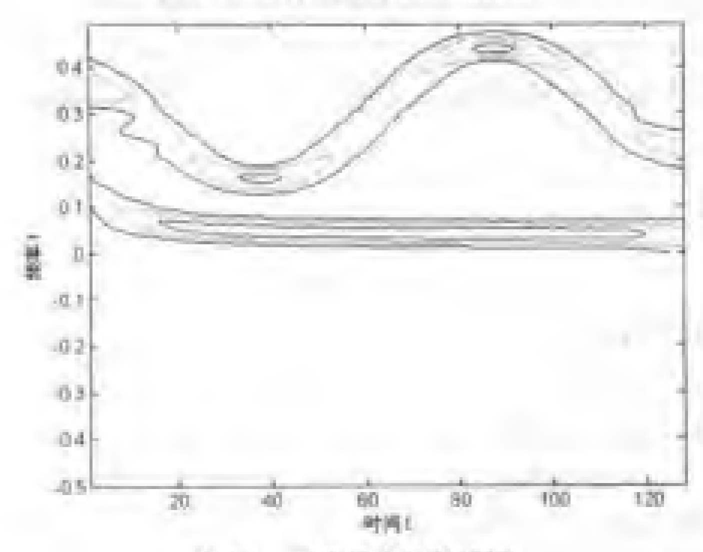


图6.4 两分量信号的谱图

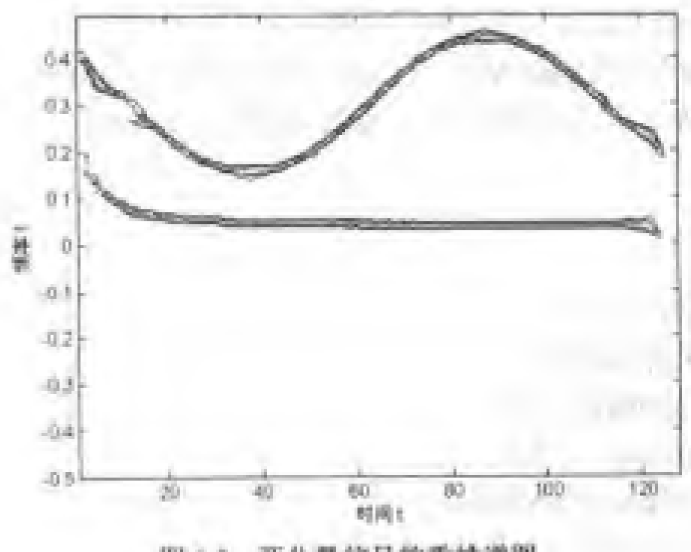


图6.5 两分量信号的重排谱图

6.1.2 Cohen 类时频分布的重排

同样，上述重排原理可以直接用于其他的分布。实际上，Cohen 类时频分布可以看成 Wigner-Ville 分布的二维卷积，即

$$C_x(t,f;\Pi)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\Pi(t-s,f-\xi)W_x(s,\xi)dsd\xi \tag{6.1.10}$$

核函数Π定义了 Cohen 类时频分布的重排方法，即

$$\hat{t}=\frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}s\Pi_h(t-s,f-\xi)W_x(s,\xi)dsd\xi}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\Pi_h(t-s,f-\xi)W_x(s,\xi)dsd\xi} \tag{6.1.11}$$

$$\hat{f}=\frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\xi\Pi_h(t-s,f-\xi)W_x(s,\xi)dsd\xi}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\Pi_h(t-s,f-\xi)W_x(s,\xi)dsd\xi} \tag{6.1.12}$$

$$C_x^{(r)}(t',f';\Pi)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}C_x(t,f;\Pi)\delta(t'-\hat{t})\delta(f'-\hat{f})dtdf \tag{6.1.13}$$

上述得到的重排分布使用了一个很好的自适应平滑核函数，同时增加了信号的时频聚集性，能有效地降低交叉项的干扰，不过重排后的 Cohen 类时频分布不再是双线性的。时频分析工具箱中计算典型重排 Cohen 类时频分布的函数有：tfrrspwv.m、tfrrpmh.m 以及 tfrrppag.m。

1. tfrrspwv 函数

【语法格式】

① [tfr,rtfr,hat]= tfrrspwv (x)

② [tfr,rtfr,hat]= tfrrspwv (x,t)

③ [tfr,rtfr,hat]= tfrrspwv (x,t,N)

④ [tfr,rtfr,hat]= tfrrspwv (x,t,N,h)

⑤ [tfr,rtfr,hat]= tfrrspwv (x,t,N,h,trace)

【使用说明】计算伪 Wigner-Ville 分布和它重排后的结果。

输入参数：x 是分析的信号；N 是频率点数；h 是频率平滑窗。

输出参数：tfr 返回伪 Wigner-Ville 分布值；rtfr 返回重排伪 Wigner-Ville 分布值；hat 返回重排向量的复数矩阵。当没有输出参数时调用函数 tfrqview.m。

2. tfrrppag 函数

【语法格式】

① [tfr,rtfr,hat]= tfrrppag (x)

② [tfr,rtfr,hat]= tfrrppag (x,t)

③ [tfr,rtfr,hat]= tfrrppag (x,t,N)

④ [tfr,rtfr,hat]= tfrrppag (x,t,N,h)

⑤ [tfr,rtfr,hat]= tfrrppag (x,t,N,h,trace)

【使用说明】计算伪 Page 分布和它重排后的结果。

输入参数：x 是分析的信号；N 是频率点数；h 是频率平滑窗。

输出参数：tfr 返回伪 Page 分布值；rtfr 返回重排伪 Page 分布值；hat 返回重排向量的复数矩阵。当没有输出参数时调用函数 tfrqview.m。

3. tfrrpmh 函数

【语法格式】

① [tfr,rtfr,hat]= tfrrpmh (x)

② [tfr,rtfr,hat]= tfrrpmh (x,t)

③ [tfr,rtfr,hat]= tfrrpmh (x,t,N)

④ [tfr,rtfr,hat]= tfrrpmh (x,t,N,h)

⑤ [tfr,rtfr,hat]= tfrrpmh (x,t,N,h,trace)

【使用说明】计算伪 Margenau-Hill 分布和它重排后的结果。

输入参数：x 是分析的信号；N 是频率点数；h 是频率平滑窗。

输出参数：tfr 返回伪 Margenau-Hill 分布值；rtfr 返回重排伪 Margenau-Hill 分布值；hat 返回重排向量的复数矩阵。当没有输出参数时调用函数 tfrqview.m。

例程 6-3：多分量信号的几类 Cohen 类时频分布与重排后的结果比较。

f6_3.m

```
%产生三分量信号
N=60;
%%信号分量 1
[sig1,ifl1]=fmsin(N,0.15,0.35,50,1,0.35,1);
%%信号分量 2
[sig2,ifl2]=fmlin(N,0.3,0.1);
%%信号分量 3
[sig3,ifl3]=fmconst(N,0.4);
sig=[sig1;zeros(8,1);sig2+sig3];
%时域波形
figure(1);
plot(real(sig),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
%获取坐标
[tfr,t,f]=tfrwv(sig);
%Wigner-Ville 分布
figure(2);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
[tfr,rtfr,hat]=tfrrspwv(sig);
%重排平滑伪 Wigner-Ville 分布
```

```
figure(3);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
%计算重排谱图
[tfr,rtfr,hat]=tfrrsp(sig);
figure(4)
%显示重排谱图
contour(t,2*f,rtfr);
xlabel('时间 t');
ylabel('频率 f');
%计算重排 M-H 分布
[tfr,rtfr,hat]=tfrrpmh(sig);
figure(5)
%显示重排 M-H 分布
contour(t,2*f,rtfr);
xlabel('时间 t');
ylabel('频率 f');
%计算重排 Page 分布
[tfr,rtfr,hat]=tfrrppag(sig);
figure(6)
%显示重排 Page 分布
contour(t,2*f,rtfr);
xlabel('时间 t');
ylabel('频率 f');
```

产生的三分量信号波形如图 6.6 所示，其 Wigner-Ville 分布如图 6.7 所示、重排平滑伪 Wigner-Ville 分布如图 6.8 所示、重排谱图如图 6.9 所示、重排 M-H 分布如图 6.10 所示以及重排 Page 分布如图 6.11 所示。

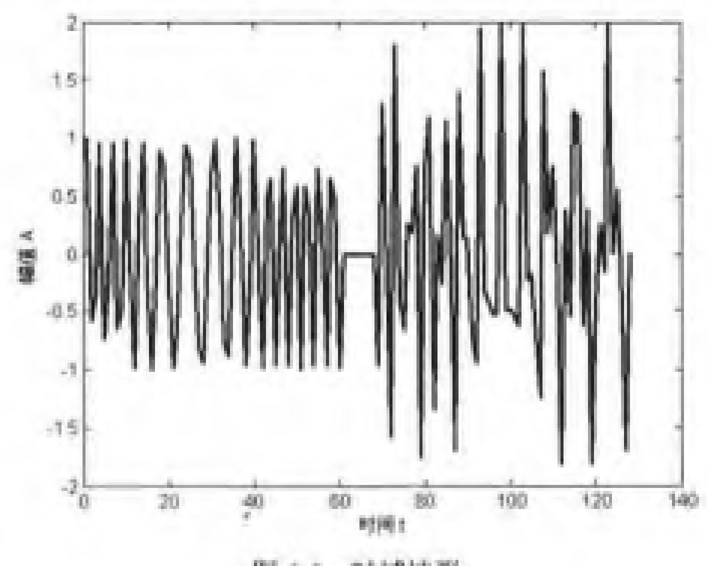


图 6.6　时域波形

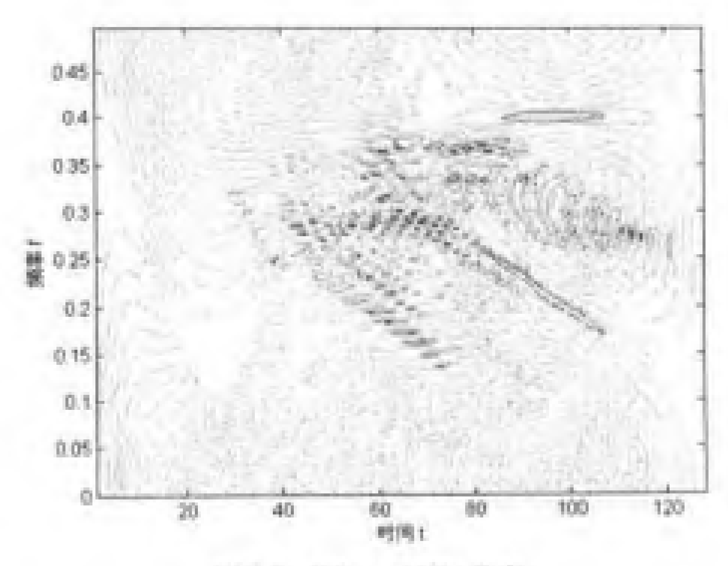


图 6.7 Wigner-Ville 分布

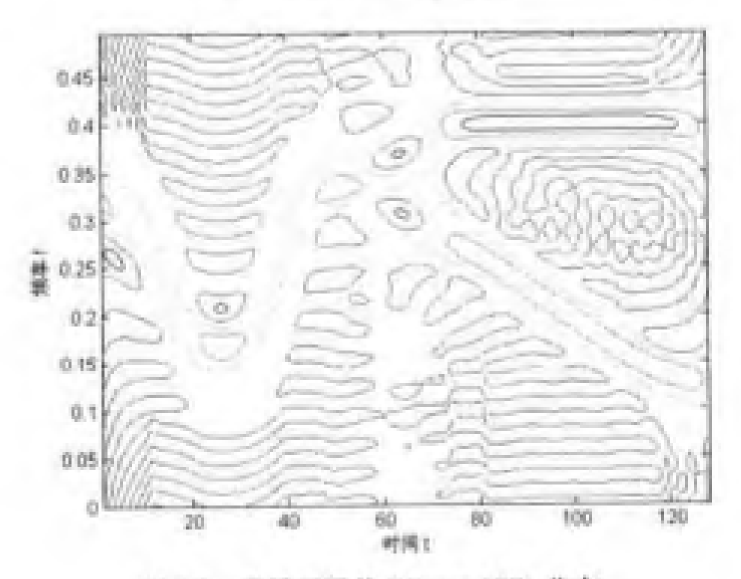


图 6.8 重排平滑伪 Wigner-Ville 分布

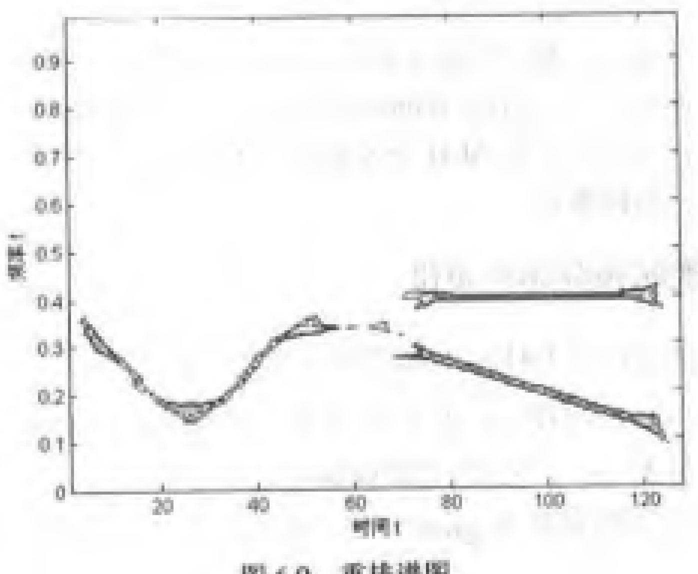


图 6.9 重排谱图

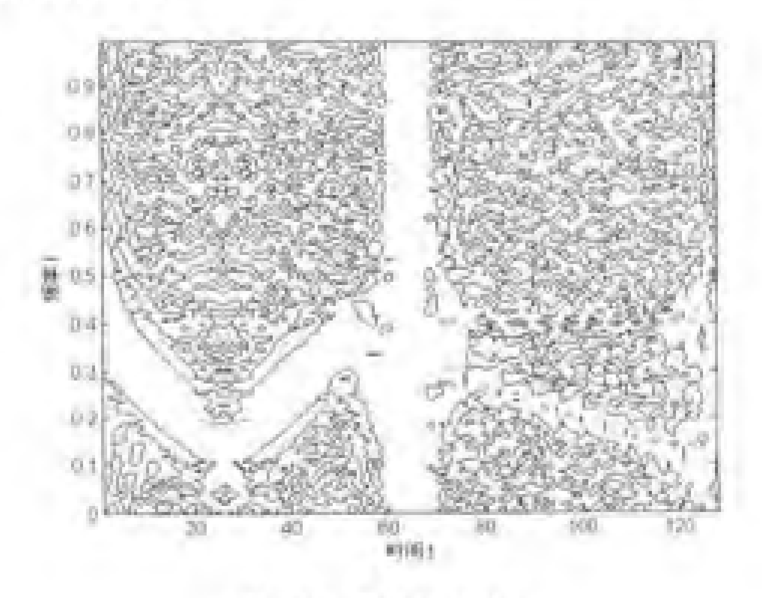


图 6.10　重排 M-H 分布

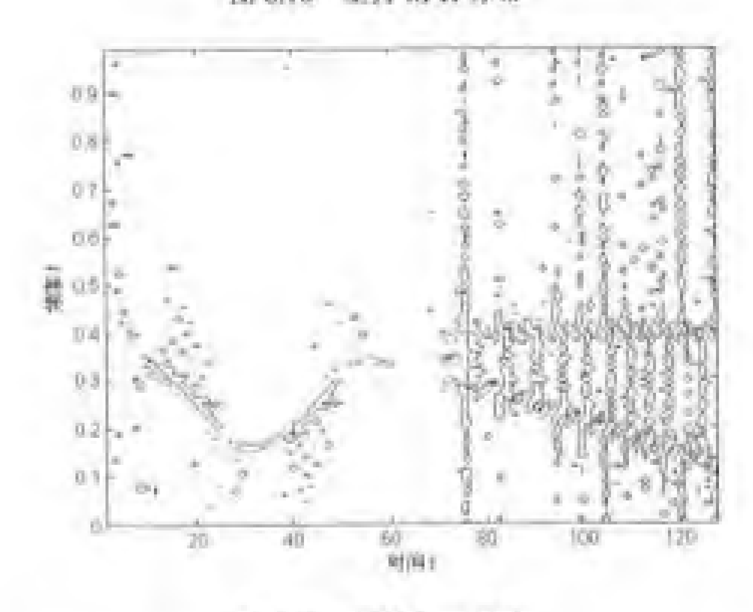


图 6.11　重排 Page 分布

在 Wigner-Ville 分布中，每个信号分量具有很好的聚集性，但是存在大量的交叉项，使得时频面难以解释；而在重排平滑伪 Wigner-Ville 分布中，则明显抑制了交叉项，但是信号分量的聚集性变差了；另外在重排 M-H 分布和重排 Page 分布中也较好地抑制了交叉项，但是时频聚集性以重排的谱图最好。

6.1.3　Affine 类时频分布的重排

同样地，重排方法可以用于时间-尺度能量分布，根据下列定义：

$$\Omega_x(t,a;\Pi)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\Pi(s/a,f_0-a\xi)W_x(t-s,\xi)dsd\xi \qquad (6.1.14)$$

可以看出，在任何点（t, $a=f_0/f$）处的时频表示等于位于以（t, f）为中心、以Π为边界的区域中的点集（$t-s$, ξ）的加权 Wigner-Ville 分布的均值。由此可以定义 Affine 类时频分布的重排方法：

$$\hat{t} = t - \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s\Pi(s/a, f_0 - a\xi)W_x(t-s,\xi)dsd\xi}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Pi(s/a, f_0 - a\xi)W_x(t-s,\xi)dsd\xi} \tag{6.1.15}$$

$$\hat{f} = \frac{f_0}{\hat{a}} = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \xi\Pi(s/a, f_0 - a\xi)W_x(t-s,\xi)dsd\xi}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Pi(s/a, f_0 - a\xi)W_x(t-s,\xi)dsd\xi} \tag{6.1.16}$$

$$\Omega_x^{(r)}(t', a'; \Pi) = \frac{1}{a'^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} a^2(t,a;\Pi)\delta(t'-\hat{t})\delta(a'-\hat{a})dtda \tag{6.1.17}$$

时频分析工具箱中有计算 Morlet 小波的尺度图函数 tfrrmsc.m。

tfrrmsc 函数

【语法格式】

① [tfr,rtfr,hat]= tfrrmsc (x)

② [tfr,rtfr,hat]= tfrrmsc (x,t)

③ [tfr,rtfr,hat]= tfrrmsc (x,t,N)

④ [tfr,rtfr,hat]= tfrrmsc (x,t,N,f0t)

⑤ [tfr,rtfr,hat]= tfrrmsc (x,t,N,h,f0t,trace)

【使用说明】计算 Morlet 小波尺度图和它重排后的结果。

输入参数：x 是分析的信号；N 是频率点数；f0t 是母小波的时宽乘积。

输出参数：tfr 返回谱图值；rtfr 返回重排谱图值；hat 返回重排向量的复数矩阵。当没有输出参数时调用函数 tfrqview.m。

例程 6-4：计算线性调频信号的重排 Morlet 小波尺度图。

f6_4.m

```
%产生线性调频信号
sig=fmlin(64,0.1,0.4);
%获取坐标
[tfr,t,f]=tfrsp(sig);
%计算重排 Morlet 小波尺度图
[tfr,rtfr,hat]=tfrrmsc(sig,1:64,64,2.1,1);
%显示 Morlet 小波尺度图
figure(1);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
figure(2)
%显示重排 Morlet 小波尺度图
contour(t,f,rtfr);
xlabel('时间 t');
ylabel('频率 f');
```

计算得到线性调频信号的 Morlet 小波尺度图如图 6.12 所示，其重排后的结果如图 6.13

所示，比较可见重排后的尺度图具有更好的时频聚集性。

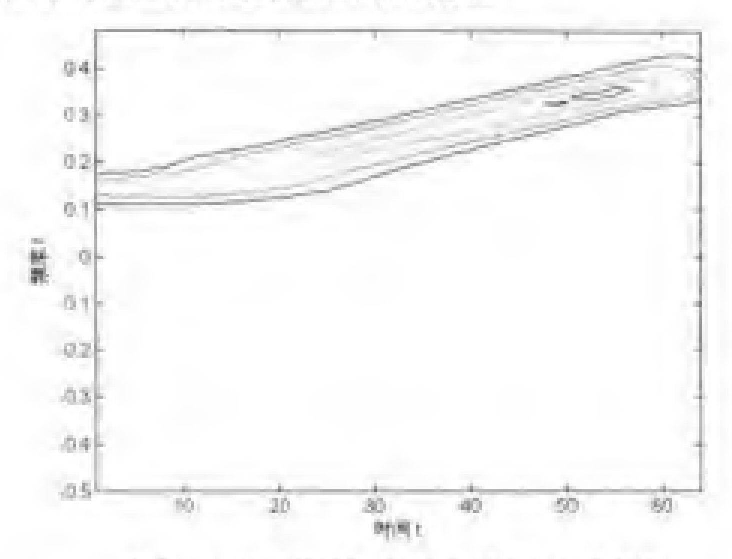


图 6.12　线性调频信号的 Morlet 小波尺度图

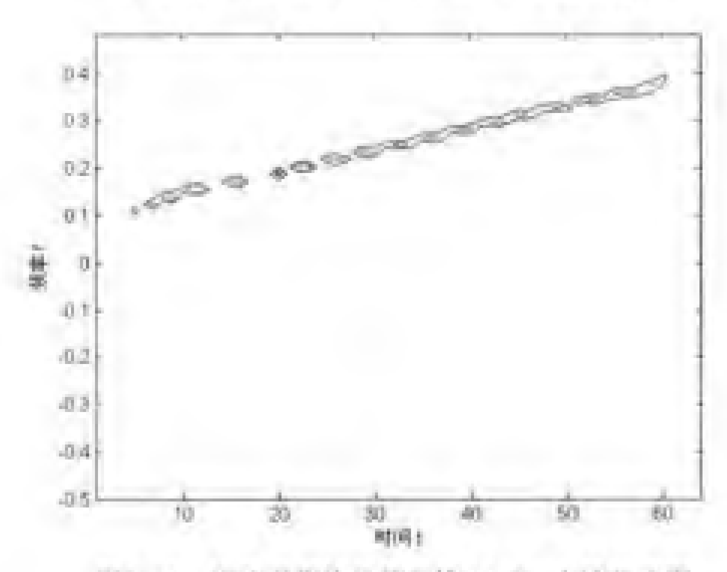


图 6.13　线性调频信号的重排 Morlet 小波尺度图

6.2　时频图像的信息提取

在前面我们已经介绍了在时频面表示非平稳信号的各种方法，接下来考虑描述信号频率随时间变化的时频图像的解释问题，而且不同的时频表示有不同的时频图像解释方法。

6.2.1　矩和边缘特性

对有些时频表示来说，矩和边缘特性提供了信号的重要信息，比如为了解调信号，它的幅值调制或瞬时频率就是重要的参数。

一、矩

一个时频能量分布 *tfr* 的一阶和二阶时间矩分别定义为：

$$f_m(t)=\frac{\int_{-\infty}^{\infty} f\cdot tfr(t,f)df}{\int_{-\infty}^{\infty} tfr(t,f)df} \tag{6.2.1}$$

$$B^2(t)=\frac{\int_{-\infty}^{\infty} f^2\cdot tfr(t,f)df}{\int_{-\infty}^{\infty} tfr(t,f)df}-f_m(t)^2 \tag{6.2.2}$$

一个时频能量分布 *tfr* 的一阶和二阶频率矩分别定义为：

$$t_m(f)=\frac{\int_{-\infty}^{\infty} t\cdot tfr(t,f)dt}{\int_{-\infty}^{\infty} tfr(t,f)dt} \tag{6.2.3}$$

$$T^2(t)=\frac{\int_{-\infty}^{\infty} t^2\cdot tfr(t,f)dt}{\int_{-\infty}^{\infty} tfr(t,f)dt}-t_m(f)^2 \tag{6.2.4}$$

它们分别定义了信号的平均位置以及在时间和频率上的跨度。例如对某个特殊的分布来说，信号的一阶时间矩对应了它的瞬时频率，而一阶频率矩对应了信号的群延迟。

时频分析工具箱中计算时间矩的函数为 momttfr.m，计算频率矩的函数为 momftfr.m。

1. momttfr 函数

【语法格式】

① [fm,B2]= momttfr(tfr,method)

② [fm,B2]= momttfr(tfr,method,fbmin)

③ [fm,B2]= momttfr(tfr,method,fbmin,fbmax)

④ [fm,B2]= momttfr(tfr,method,fbmin,fbmax,freqs)

【使用说明】计算时频表示的一阶和二阶时间矩

输入参数：tfr 是时频表示（$N\times M$）；method 是选择的表示方法；fbmin 是最小频率点（1）；fbmax 是最大频率点（M）；freqs 是每个频率点对应的归一化频率值。

输出参数：fm 返回一阶矩；B2 返回二阶矩。

例程 6-5：计算线性调频信号的时间矩。

f6_5.m

```
%产生线性调频信号
sig=fmlin(200,0.1,0.4);
%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig);
%计算一阶和二阶矩
[fm,B2]=momttfr(tfr,'tfrwv');

%结果显示
figure(1);
plot(fm,'LineWidth',2);
```

```
xlabel('时间 t');
ylabel('一阶矩 fm');
figure(2);
plot(B2,'LineWidth',2);
xlabel('时间 t');
ylabel('二阶矩 B2');
```

计算得到线性调频信号的一阶时间矩如图 6.14 所示，其二阶时间矩如图 6.15 所示。

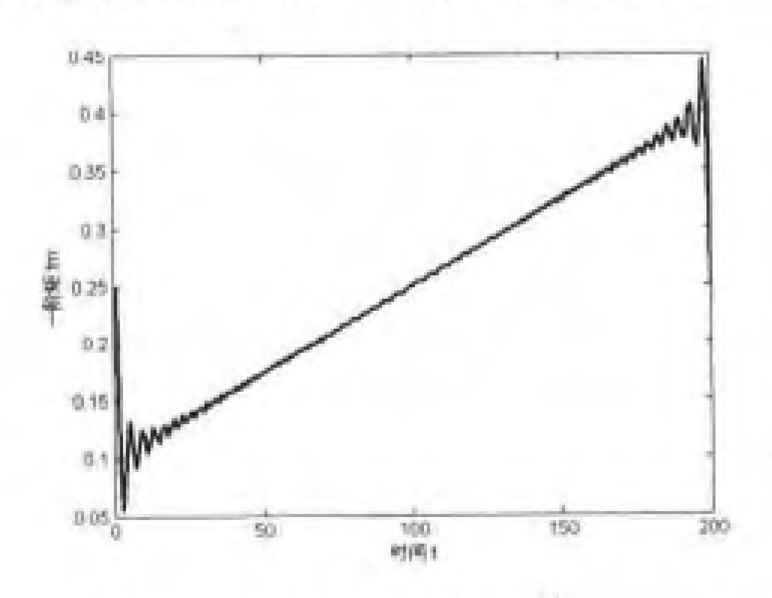


图 6.14　一阶时间矩

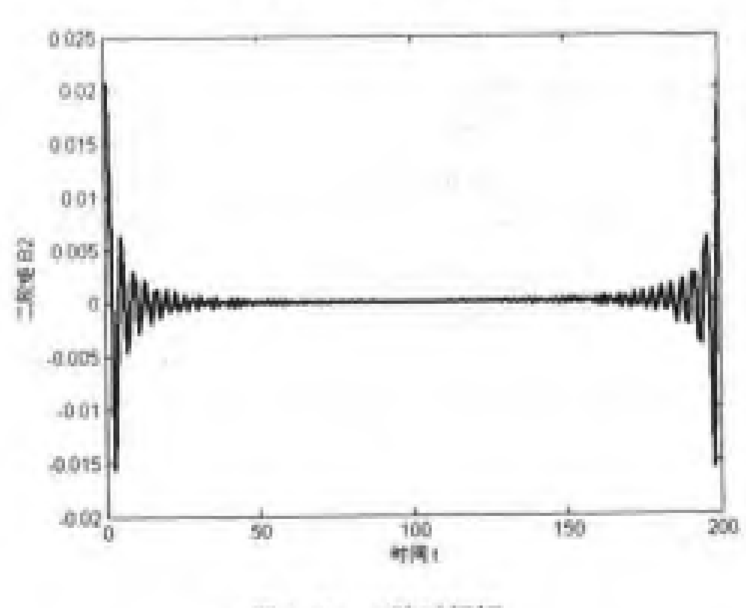


图 6.15　二阶时间矩

2. momftfr 函数

【语法格式】

① [tm,T2]= momftfr (tfr)

② [tm,T2]= momftfr (tfr,tmin)

③ [tm,T2]= momftfr (tfr,tmin,tmax)

④ [tm,T2]= momftfr (tfr,tmin,tmax,time)

【使用说明】计算时频表示的一阶和二阶频率矩。

输入参数：tfr 是时频表示（$N\times M$）；tmin 是最小列元素（1）；tmax 是最大列元素（M）；time 是时间点（1:M）。

输出参数：tm 返回一阶矩；T2 返回二阶矩。

例程 6-6：计算线性调频信号的频率矩。

f6_6.m

```
%产生线性调频信号
sig=fmlin(200,0.1,0.4);
%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig);
%计算一阶和二阶矩
[tm,T2]=momftfr(tfr);

%结果显示
figure(1);
plot(f,tm,'LineWidth',2);
xlabel('频率 f');
ylabel('一阶矩 tm');
figure(2);
plot(f,T2,'LineWidth',2);
xlabel('频率 f');
ylabel('二阶矩 T2');
```

计算得到线性调频信号的一阶频率矩如图 6.16 所示，其二阶频率矩如图 6.17 所示。

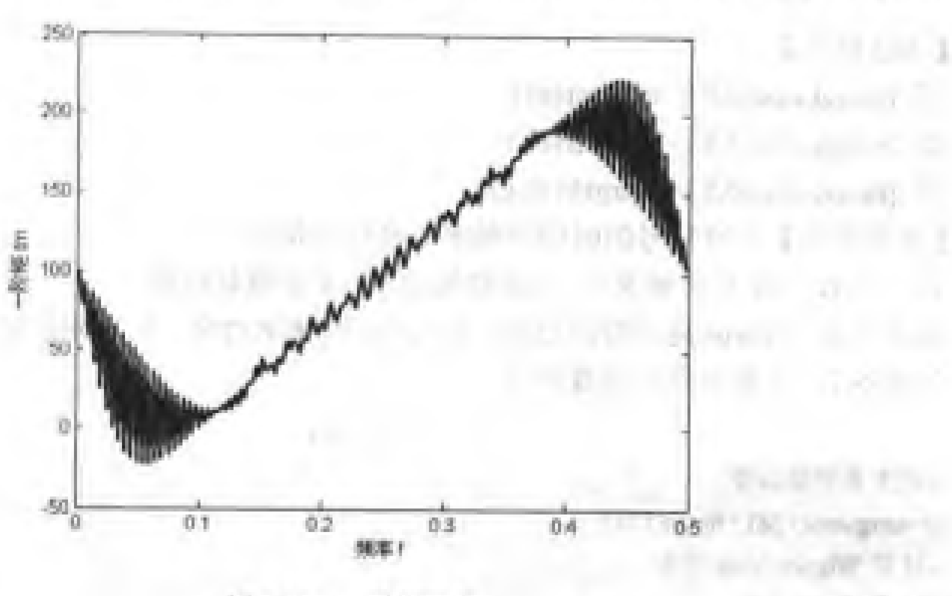


图 6.16 一阶频率矩

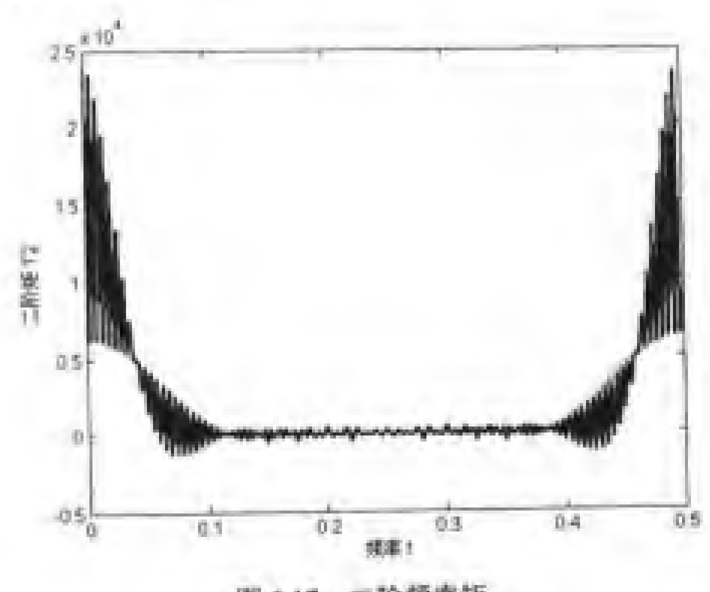


图 6.17　二阶频率矩

二、边缘特性

一个时频表示的边缘分布也是重要的参数，时间边缘定义为：

$$m_f(t)=\int_{-\infty}^{\infty} tfr(t,f)df \tag{6.2.5}$$

频率边缘定义为：

$$m_t(f)=\int_{-\infty}^{\infty} tfr(t,f)dt \tag{6.2.6}$$

从物理意义上来说，一个时频分布的时间边缘对应于信号的瞬时能量，而频率边缘对应于能量谱密度，即

$$m_f(t)=|x(t)|^2, m_t(f)=|X(f)|^2 \tag{6.2.7}$$

时频分析工具箱中计算信号边缘特性的函数为 margtfr.m。

margtfr 函数

【语法格式】

① [margt,margf,E]= margtfr(tfr)

② [margt,margf,E]= margtfr(tfr,t)

③ [margt,margf,E]= margtfr(tfr,t,f)

【使用说明】计算信号的时间和频率边缘以及能量。

输入参数：tfr 是时频表示；t 是时间向量；f 是频率向量。

输出参数：margt 返回时间边缘；margf 返回频率边缘；E 返回能量。

例程 6-7：计算信号的边缘特性。

f6_7.m

```
%产生非平稳信号
sig=amgauss(128).*fmlin(128);
%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig);
```

```
%计算边缘特性
[margt,margf,E]=margtfr(tfr);
%结果显示
figure(1);
plot(t,margt,'LineWidth',2);
xlabel('频率 f');
ylabel('时间边缘 m(t)');
figure(2);
plot(f,margf,'LineWidth',2);
xlabel('频率 f');
ylabel('频率边缘 m(f)');
E
```

计算得到信号的时间边缘特性如图 6.18 所示，频率边缘特性如图 6.19 所示，信号能量为 E = 16.0000。

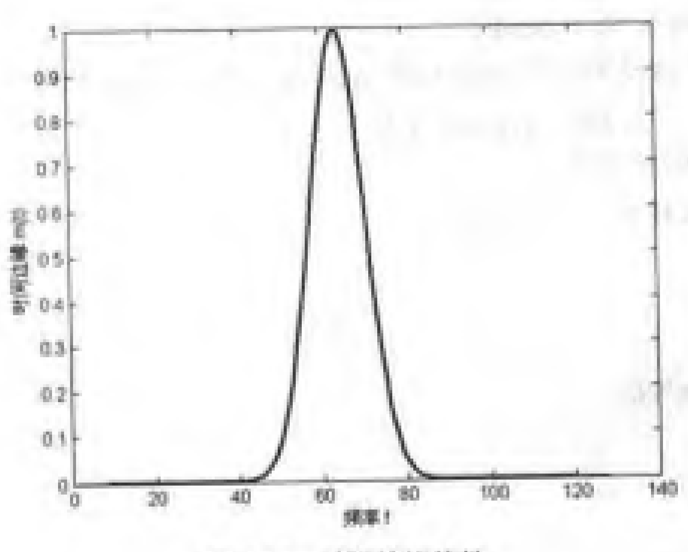


图 6.18 时间边缘特性

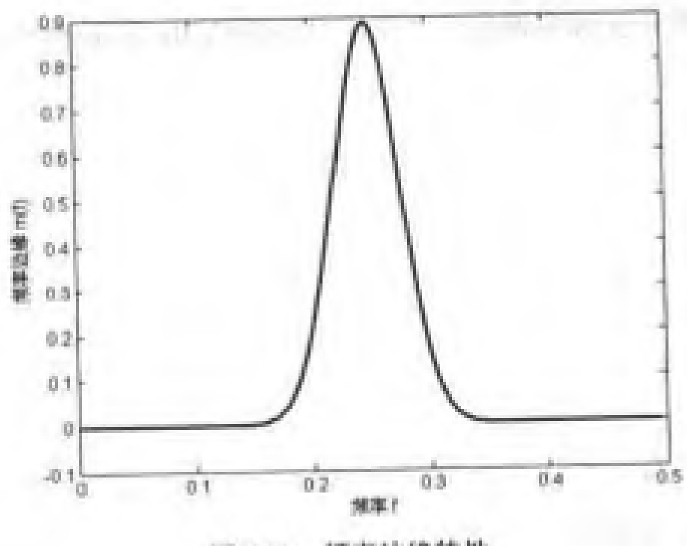


图 6.19 频率边缘特性

6.2.2 干扰项中的相位信息

任何双线性时频分布中存在固有的交叉项干扰，虽然它增加了信号时频面解释的难度，但是它也包含了解析信号的信息。干扰项的结构和构成规则有利于说明它们包含的信息。例如干扰项包含了信号的相位信息，为了说明这一点，时频分析工具箱提供了两个函数：movpwdph.m 和 movpwjph.m。

movpwdph 计算由两个具有不同相移的常频率调制信号叠加后的非平稳信号的伪 Wigner-Ville 分布，通过比较不同的相移来观察振荡交叉项在时间轴上的滑动。

1. movpwdph 函数

【语法格式】

① M= movpwdph(N)

② M= movpwdph(N,Np)

③ M= movpwdph(N,Np,typesig)

【使用说明】产生移动帧说明相移对伪 Wigner-Ville 分布交叉项的影响。

输入参数：N 是信号点数；typesig 是信号类型。

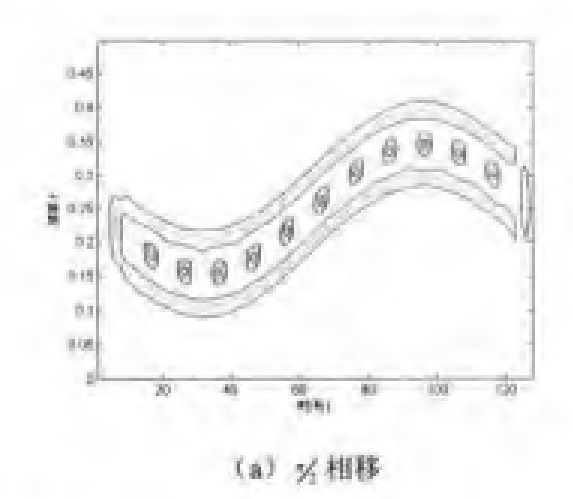
输出参数：M 返回帧矩阵。

例程 6-8：分析相移对交叉项的影响。

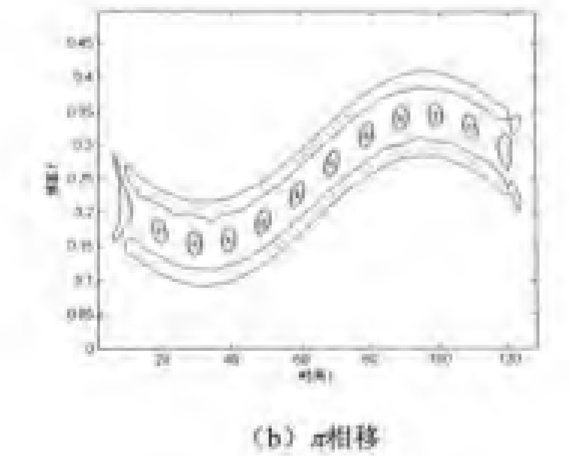
f6_6.m

```
Np=8;
M=movpwdph(128,Np,'S');
for i=2:2:Np
    movie(M(:,i));
    pause;
end
```

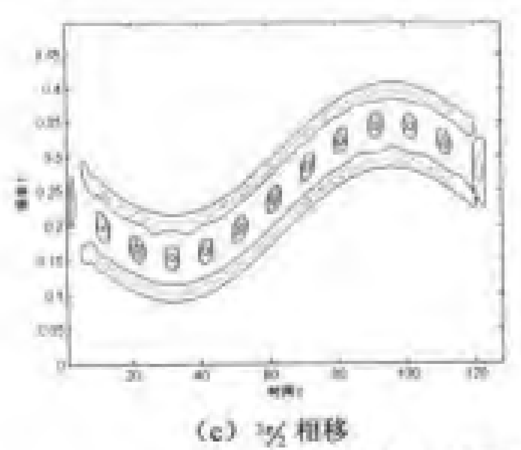
计算不同相移时两正弦频率调制信号的伪 Wigner-Ville 分布如图 6.20 所示。



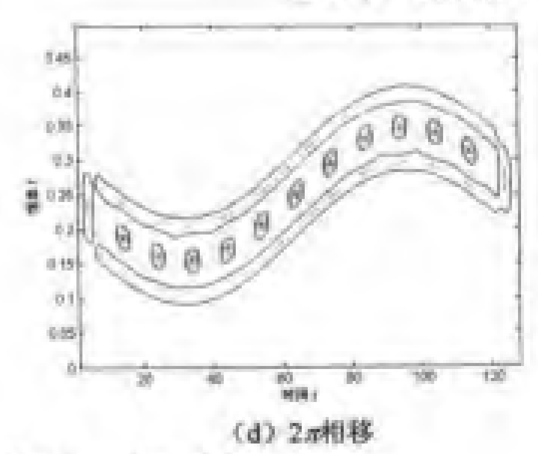
（a）$\pi/2$ 相移



（b）π相移



（c）$^{3\pi}/_{2}$ 相移



（d）2π相移

图 6.20　不同相移时的伪 Wigner-Ville 分布

函数 movpwjph 计算信号相位的跳跃对伪 Wigner-Ville 分布交叉项的影响。

2. movpwjph 函数

【语法格式】

① M= movpwjph (N)

② M= movpwjph (N,Np)

③ M= movpwjph (N,Np,typesig)

【使用说明】产生移动帧说明相移对伪 Wigner-Ville 分布交叉项的影响。

输入参数：N 是信号点数；typesig 是信号类型。

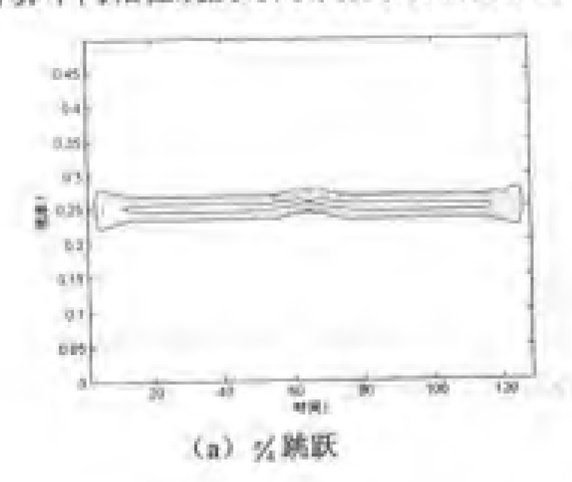
输出参数：M 返回帧矩阵。

例程 6-9：分析相位跳跃对交叉项的影响。

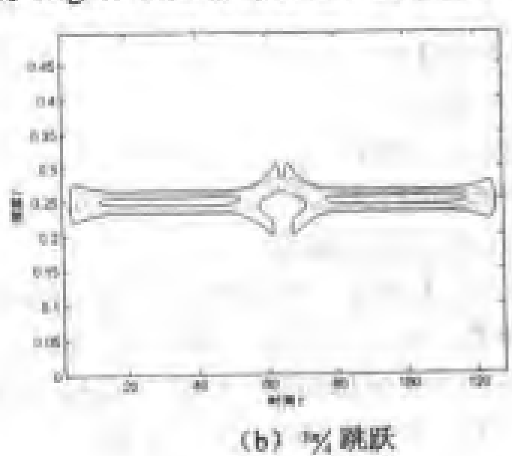
f6_9.m

```
Np=8;
M=movpwjph(128,Np,'C');
for i=1:2:Np-1
    movie(M(:,i));
    pause;
end
```

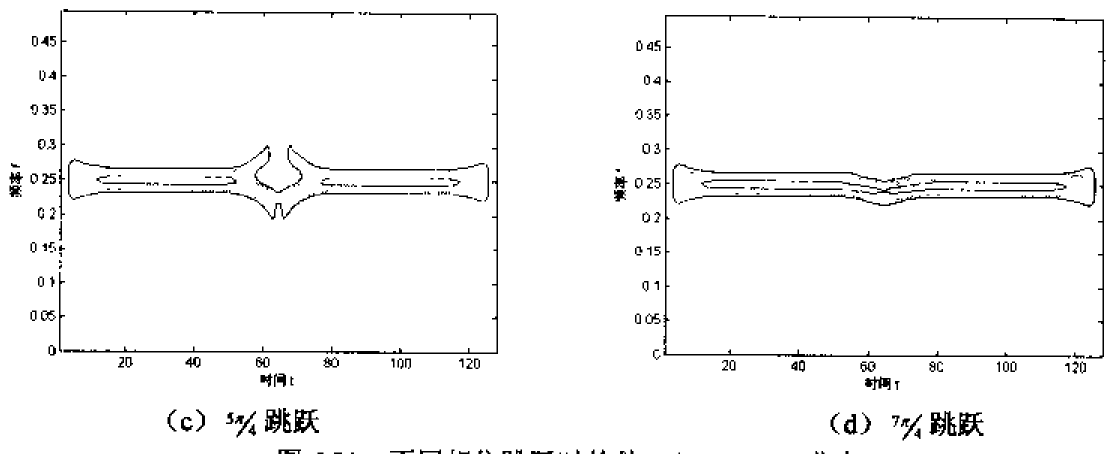
计算不同相位跳跃时两常频率调制信号的伪 Wigner-Ville 分布如图 6.21 所示。



（a）$^{\pi}/_{4}$ 跳跃



（b）$^{3\pi}/_{4}$ 跳跃



（c）$^{5\pi}/_4$ 跳跃　　（d）$^{7\pi}/_4$ 跳跃

图 6.21　不同相位跳跃时的伪 Wigner-Ville 分布

6.2.3　Renyi 信息

另外一个关于非平稳信号的重要信息是基本信号分量的个数，这也将会产生另外一个问题：为了确认信号中存在两个基本分量，而不是一个，这两个基本分量的距离至少应该是多少？为了解决这个问题，人们提出了利用信号时频分布信息度量的方法。但是我们常见的 Shannon 定理，即

$$I_x = -\int_{-\infty}^{\infty} f(x)\log_2 f(x)dx \tag{6.2.8}$$

在此不能应用（其中 $f(x)$ 是 x 的概率密度函数），因为时频分布存在负值。为此人们定义了更一般的信息度量形式，它允许时频分布中的负值存在，即 Renyi 信息，定义为：

$$R_x^{\alpha} = \frac{1}{1-\alpha}\log_2\{\int_{-\infty}^{\infty} f^{\alpha}(x)dx\} \tag{6.2.9}$$

在连续情况下，α是信息阶数。一阶 Renyi 信息（α=1）就退化为 Shannon 信息。

对时频分布 $C_x(t,f)$，定义三阶 Renyi 信息为

$$R_C^3 = -\frac{1}{2}\log_2\{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} C_x^3(t,f)dtdf\} \tag{6.2.10}$$

计算结果是以位（bits）为单位的。如果一个基本分量信号产生零位信息（2^0），那么两个明显分开的基本分量信号将产生 1 位信息（2^1），4 个明显分开的基本分量信号将产生 2 位信息（2^2），如此类推。

时频分析工具箱中计算 Renyi 信息的函数为 renyi.m。

renyi 函数

【语法格式】

① R=renyi(tfr)

② R=renyi(tfr,t)

③ R=renyi(tfr,t,f)

④ R=renyi(tfr,t,f,alpha)

【使用说明】度量二维时频分布的 Renyi 信息。

输入参数：tfr 是信号的二维时频分布；t 是时频分布矩阵的时间坐标向量（1:N）；f 是时频分布矩阵的频率坐标向量（1:M）；alpha 是 Renyi 信息的阶数。

输出参数：R 返回 Renyi 信息度量值。

例程 6-10：分别计算单分量、两分量和四分量信号的 Renyi 信息。

f6_10.m

```
%产生单分量信号
sig1=atoms(128,[64,0.25,20,1]);
%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig1);
%计算 Renyi 信息
disp('单分量信号的 Renyi 信息度量为')
R1=renyi(tfr,t,f)
%产生两分量信号
sig2=atoms(128,[32,0.25,20,1;96,0.25,20,1]);
%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig2);
%计算 Renyi 信息
disp('两分量信号的 Renyi 信息度量为');
R2=renyi(tfr,t,f)
%产生四分量信号
sig3=atoms(128,[32,0.15,20,1;96,0.15,20,1;32,0.35,20,1;96,0.35,20,1]);
%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig3);
%计算 Renyi 信息
disp('四分量信号的 Renyi 信息度量为');
R3=renyi(tfr,t,f)
```

输出结果为：

```
单分量信号的 Renyi 信息度量为
R1 =
   -0.2075
两分量信号的 Renyi 信息度量为
R2 =
    0.7790
四分量信号的 Renyi 信息度量为
R3 =
    1.8029
```

这与 Renyi 信息的定义是一致的。

6.3 Wigner–Hough 变换

6.3.1 Hough 变换

Hough 变换是 Hough 在 1962 年提出的形状匹配技术，它可将被检测图像的参数曲线在参数空间中凝聚起来形成与相应曲线对应的参数峰点，从而得到图像中各个曲线的参数。对于离散的有

限图像来说，Hough 变换的思想是将所有可能的线条参数组成的参数空间量化为有限的参数表。

考虑如下的直线极坐标参数方程：

$$x\cos\theta + y\sin\theta = \rho$$

对图像 I 的每一点 (x,y)，Hough 变换会对应 (ρ,θ) 平面上的一个正弦波，其幅值等于点 (x,y) 的密度。所以对 I 中的所有点，Hough 变换对应了一系列的正弦波，而且会在 (ρ,θ) 平面上相互交叉。也就是说，Hough 变换在图像 I 上沿着直线进行积分且积分值与参数 (ρ,θ) 相关。

时频分析工具箱中实现 Hough 变换的函数为 htl.m。

htl 函数

【语法格式】

① [HT,rho,theta]=htl(IM)

② [HT,rho,theta]=htl(IM,M)

③ [HT,rho,theta]=htl(IM,M,N)

④ [HT,rho,theta]=htl(IM,M,N,trance)

【使用说明】对图像进行 Hough 变换。

输入参数：IM 是要分析的图像，大小为 (x_{max}, y_{max})；M 是径向的样本数；N 是角度向的样本数；trance 非零时显示算法过程。

输出参数：HT 返回变换后的矩阵；rho 是径向的样本序列；theta 是角度向的样本序列。

6.3.2 W-H 变换

当对信号的 Wigner-Ville 分布进行 Hough 变换时，可以得到一种新的变换，称作 Wigner-Hough 变换，简写为 W-H 变换。

例如考虑如下的信号：

$$x(t) = e^{j2\pi(v_0 t + \beta/2 t^2)} + n(t)$$

式中 $n(t)$ 为白噪声。

其 W-H 变换为：

$$WH_x(v_0,\beta) = \int_T W_x(t, v_0 + \beta t)dt = \int_{-\infty}^{\infty}\int_T x(t+\frac{\tau}{2})x^*(t-\frac{\tau}{2})e^{-j2\pi(v_0+\beta t)}dtd\tau$$

W-H 变换与 Wigner-Ville 分布相比具有两个优点：抑制噪声以及交叉项，下面将通过两个例程来说明这一点。

例程 6-11：利用 W-H 变换在低信噪比下检测线性 Chirp 信号。

f6_11.m

```
%产生线性调频信号
%信噪比为 1dB
N=64;
sig=sigmerge(fmlin(N,0,0.3),noisecg(N),1);

%时域波形
figure(1);
subplot(2,1,1);
```

```
plot(1:N,real(sig),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');

%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig);
%显示 Wigner-Ville 分布等高线
figure(2);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');

%Hough 变换
[WH,rho,theta]=htl(tfr,N,N);
figure(3);
mesh(rho,theta,WH);
xlabel('极半径 rho');
ylabel('角度 theta');
```

信噪比为 1dB 的含噪线性调频信号如图 6.22 所示，由时域波形难以看出 Chirp 信号的存在；计算其 Wigner-Ville 分布如图 6.23 所示，从图中能隐约发现 Chirp 信号，但不十分清楚；最后对 Wigner-Ville 分布进行 Hough 变换，结果如图 6.24 所示，能明显看出在（ρ,θ）平面上存在一峰值，它就对应了 Chirp 信号。

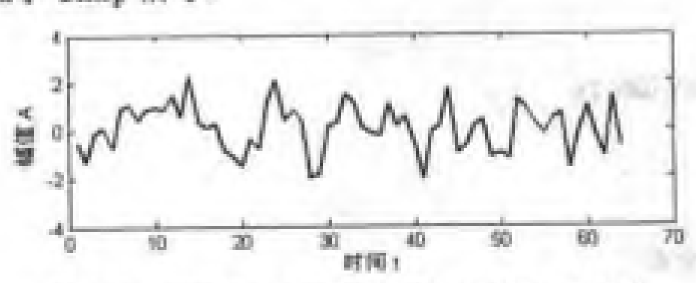


图 6.22 信噪比为 1dB 的含噪线性调频信号波形

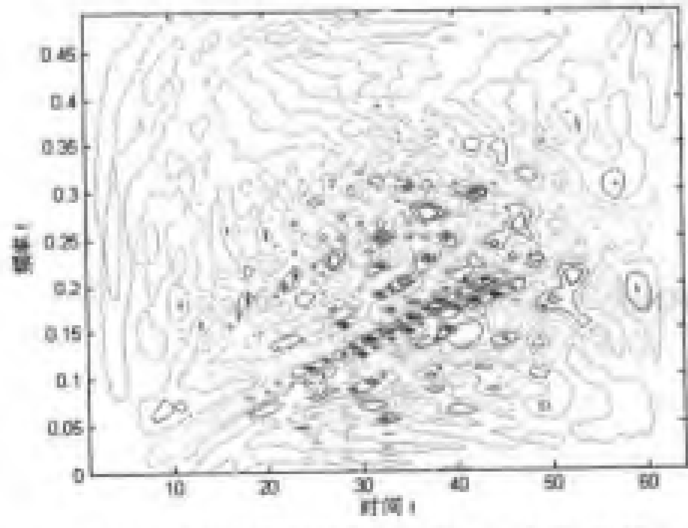


图 6.23 含噪线性调频信号的 Wigner-Ville 分布

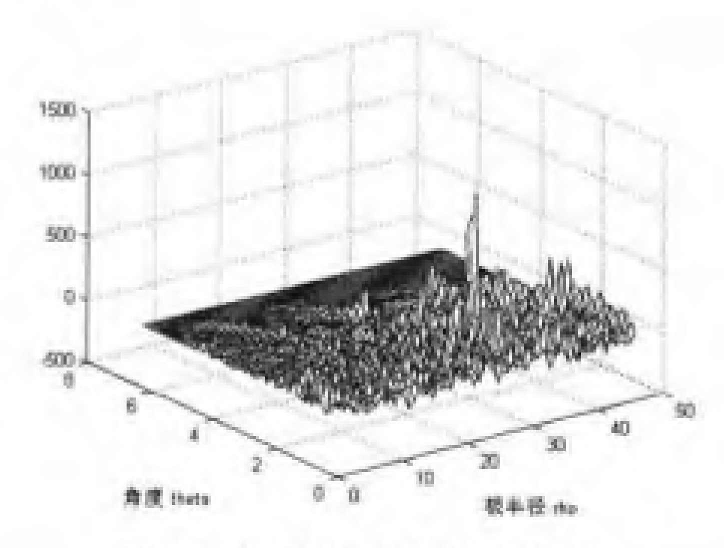


图 6.24 含噪线性调频信号的 W-H 变换

例程 6-12：比较 W-H 变换与 Wigner-Ville 分布的交叉项性能。

f6_12.m

```
%产生两分量信号
N=64;
sig=sigmerge(fmlin(N,0,0.4),fmlin(N,0.3,0.5),1);

%时域波形
figure(1);
subplot(2,1,1);
plot(1:N,real(sig),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');

%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig);
%显示 Wigner-Ville 分布等高线
figure(2);
contour(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');

%Hough 变换
[WH,rho,theta]=htl(tfr,N,N);
figure(3);
mesh(rho,theta,WH);
xlabel('极半径 rho');
ylabel('角度 theta');
```

信噪比为 1dB 的含噪线性调频信号如图 6.25 所示，计算其 Wigner-Ville 分布如图 6.26 所

示，从图中能明显看出在两个分量信号之间存在严重的交叉项；最后对 Wigner-Ville 分布进行 Hough 变换，结果如图 6.27 所示，能明显看出在（ρ,θ）平面上存在两个峰值，它们分别对应原信号中的两个信号，除此之外，（ρ,θ）平面上不存在任何其他的峰值点，这说明 WH 变换后不存在交叉项。

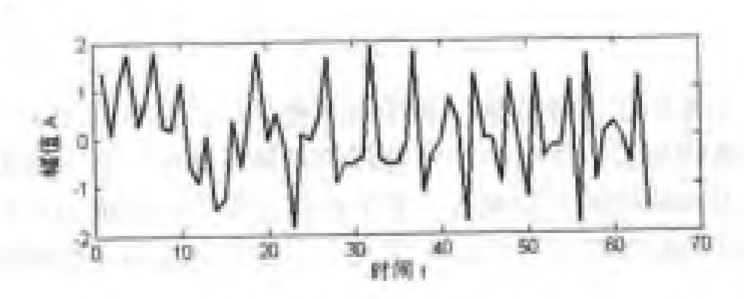


图 6.25　两分量信号的时域波形

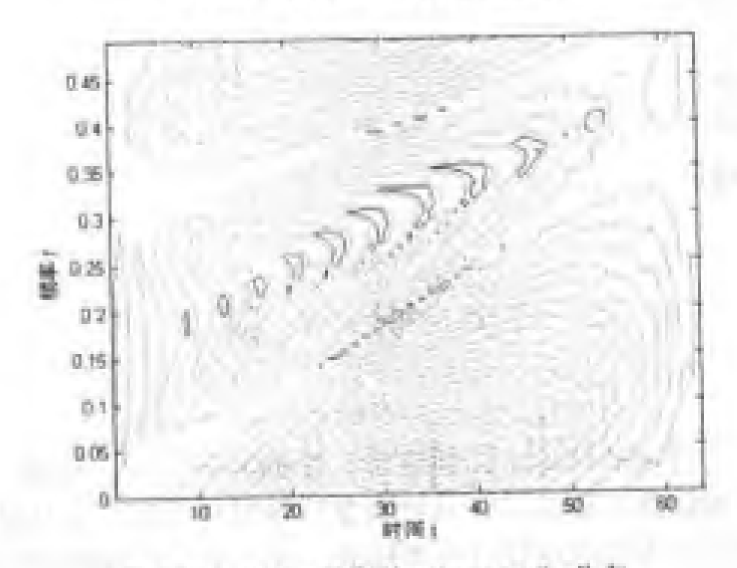


图 6.26　两分量信号的 Wigner-Ville 分布

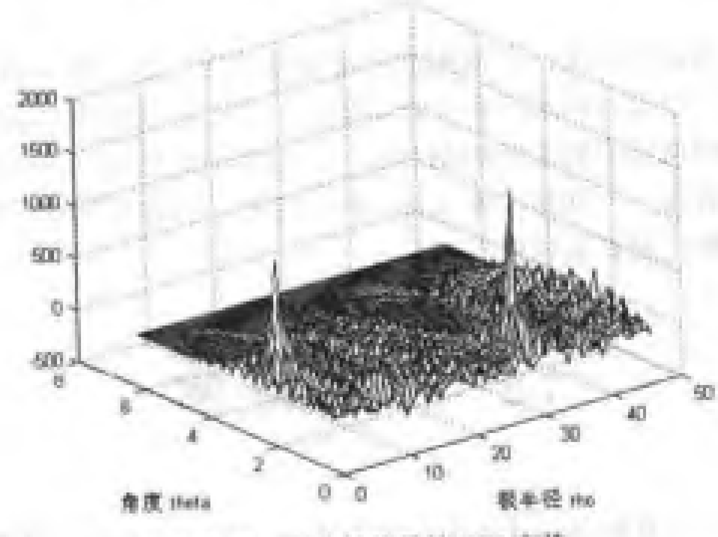


图 6.27　两分量信号的 WH 变换

第 7 章　时频分析在信号处理中的应用

在工程实际中，需要传递各种数据，其目的是把某些信息传递出去。信息的利用程度和信号处理技术的发展密切相关。传统的信号处理的对象系统限于时不变的线性系统，假设信号是平稳的。随着应用领域的扩大和深入，非平稳信号处理已经成为研究热点，广泛用于雷达、生物、医学和故障诊断等几乎所有技术领域。本章详细介绍了时频分析工具箱在几个信号处理领域中的应用。

本章主要内容包括：

- 信号检测；
- 语音信号处理；
- 医学信号处理；
- 雷达信号处理。

7.1 引　言

许多天然的和人工合成的信号，比如语音、生物医学信号、音乐、雷达和声纳信号、机械振动等，就是典型的非平稳信号，其特点是持续时间有限，并且是时变的。时频分析正是着眼于真实信号组成成分的这种时变谱特征，将一个一维的时间信号以二维的时间-频率函数的形式表示出来，旨在揭示信号中包含多少频率分量以及每个分量随时间是如何变化的。

自从时频分析技术诞生以来，其应用潜力就初露端倪。这一技术的长足进展，为许多难题的解决带来了曙光。目前，时频分析在语音识别，雷达信号处理和图像处理，地震信号处理，信号重构以及扩频通信中的干扰抑制方面，已有成功的应用。由于时频分析所涉及的应用领域众多，不可能对其一一列举，本文将对时频分析工具箱在其中一些领域中的工程应用进行说明，由此可以推广到其他领域。

7.2 信 号 检 测

信号检测是一个在人类和自然斗争中存在的永恒主题，人们基于各种目的一直在寻找和发展更加有效、更加简单的信号检测方法。时频分析作为一种新的信号处理方法，近年来在信号检测中得到越来越多的重视。

7.2.1 暂态信号检测

例程 7-1：某暂态信号是一个被单边指数幅值调制的常频率信号，它被白色高斯噪声污染，信噪比为-5dB，利用谱图来检测暂态信号。

f7_1.m

```
%产生暂态信号
transig=amexpo1s(64).*fmconst(64);
sig=[zeros(100,1);transig;zeros(92,1)];
%加噪声
sign=sigmerge(sig,noisecg(256),-5);

%绘制时域波形
figure(1);
plot(real(sign),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');

%计算功率谱
nfft=1024;
dsp=abs(fft(sign,nfft)).^2;
%绘制频谱图
figure(2);
plot((0:nfft/2-1)/nfft,dsp(1:nfft/2));
xlabel('频率 f');
ylabel('功率谱 PSD');

%计算信号的谱图
[tfr,t,f]=tfrsp(sign);
%绘图
%%绘制等高线图
figure(3);
contour(t,f(1:128),abs(tfr(1:128,:)).^2);
xlabel('时间 t');
ylabel('频率 f');
%%三维图

figure(4);
mesh(t,f(1:128),abs(tfr(1:128,:)).^2);
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

含噪的暂态信号的时域波形如图 7.1 所示，其功率谱如图 7.2 所示，由此两图可以发现难以在

时域和频域精确定位暂态信号的位置。接着计算了信号的谱图，其等高线如图 7.3 所示，三维视图如图 7.4 所示，可以明显发现，暂态信号以归一化频率 0.25 为中心，位于时间点 125 和 160 之间。

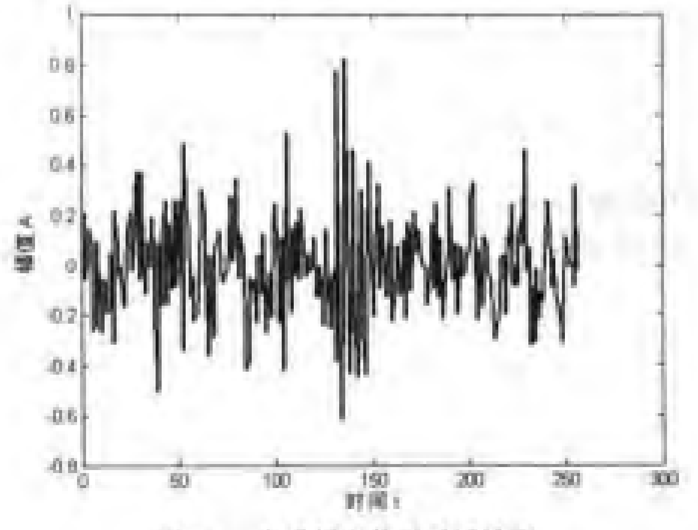


图 7.1　含噪暂态信号时域波形

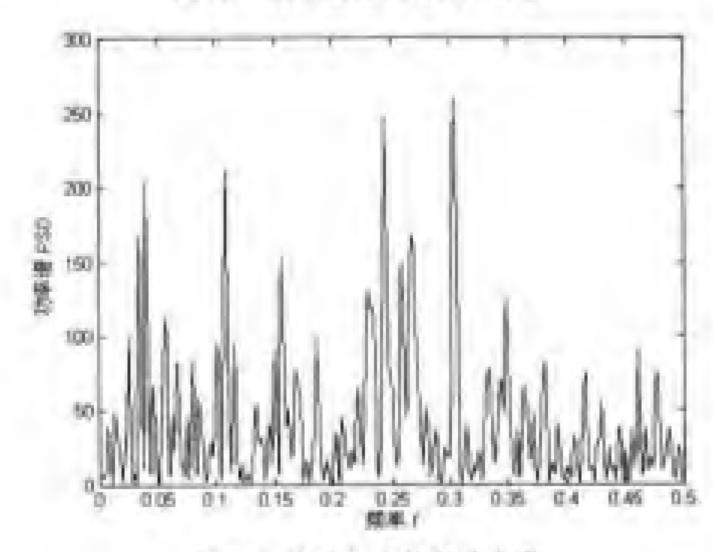


图 7.2　含噪暂态信号功率谱

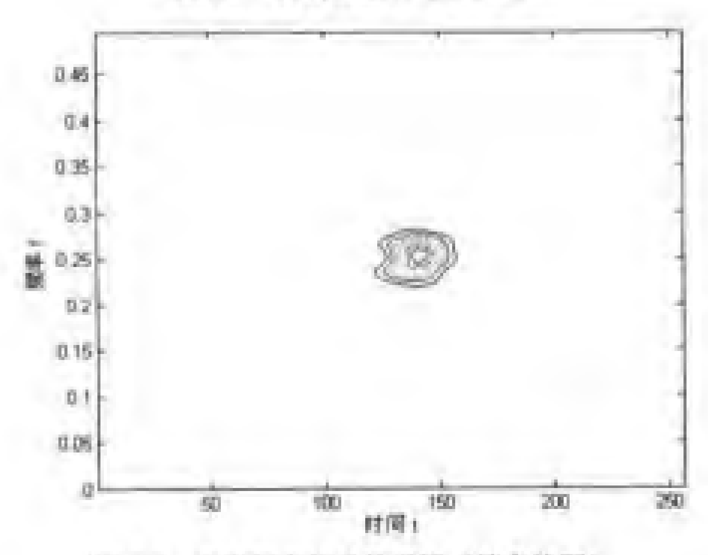


图 7.3　含噪暂态信号的谱图（等高线图）

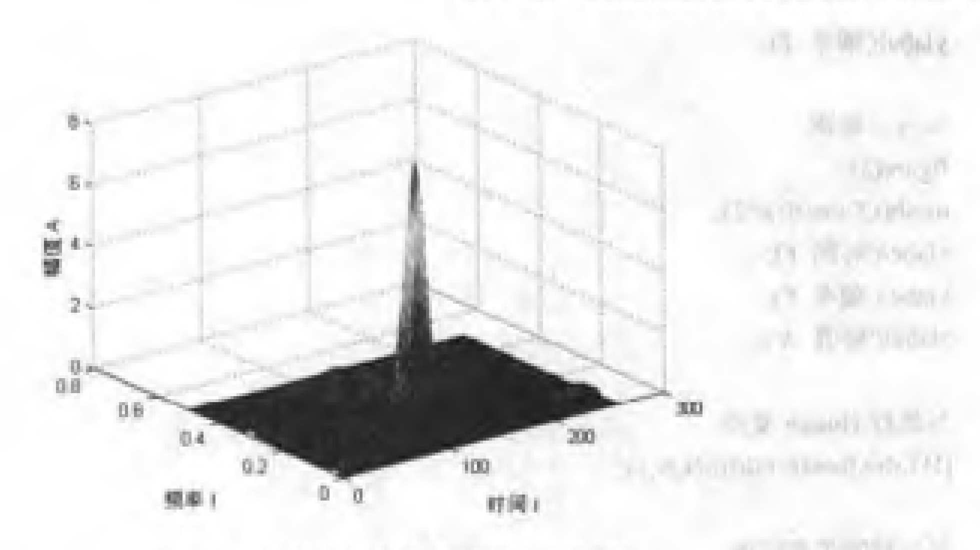


图 7.4 含噪暂态信号的谱图（三维图）

7.2.2 线性调频信号的检测和估计

我们已经知道，Wigner-Ville 分布对线性调频信号具有理想的时频聚集性（一条直线），因此检测和估计线性调频信号可以转化为检测和估计图像中的一条线，这样的问题是模式识别中的一般问题，容易解决，这可以通过对 Wigner-Ville 分布进行 Hough 变换来实现。

例程 7-2：利用 Hough 变换检测单线性调频信号（信噪比为 1dB）。

f7_2.m

```
%产生线性调频信号
N=64;
sig=fmlin(N,0,0.3);
%加噪声
sign=sigmerge(sig,noisecg(N),1);

%绘制时域波形
figure(1);
plot(real(sign),'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');

%计算信号的 Wibner-Ville 分布
[tfr,t,f]=tfrwv(sign);

%绘图
%%绘制等高线图
figure(2);
contour(t,f,abs(tfr).^2);
xlabel('时间 t');
```

```
ylabel('频率 f');

%%三维图
figure(3);
mesh(t,f,abs(tfr).^2);
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');

%进行 Hough 变换
[HT,rho,theta]=htl(tfr,N,N,1);

%%绘制等高线图
figure(4);
contour(rho,theta,HT');
xlabel('变量 rho');
ylabel('变量 theta');

%%三维图
figure(5);
mesh(rho,theta,HT');
xlabel('变量 rho');
ylabel('变量 theta');
zlabel('幅值 A');
```

单线性调频信号的时域波形如图 7.5 所示。单线性调频信号的 Wigner-Ville 分布的等高线图如图 7.6 所示、三维图如图 7.7 所示。接着计算其 Wigner-Hough 变换如图 7.8（等高线图）和图 7.9（三维图）所示，可以看出在（ρ,θ）平面上存在一个峰值，它表征了线性调频信号。

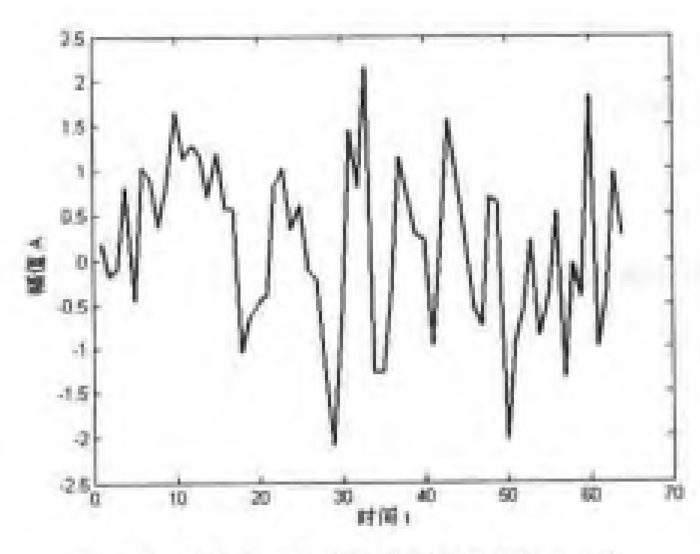


图 7.5　单线性调频信号波形（信噪比 1dB）

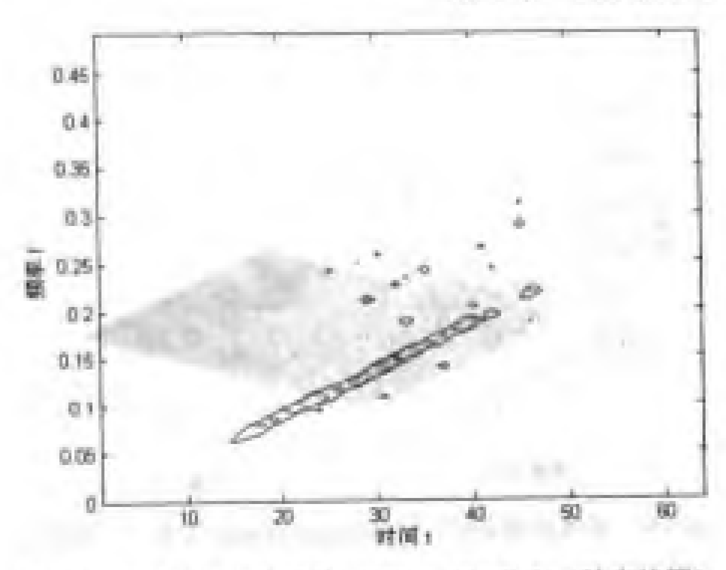


图 7.6　单线性调频信号的 Wigner-ville 分布（等高线图）

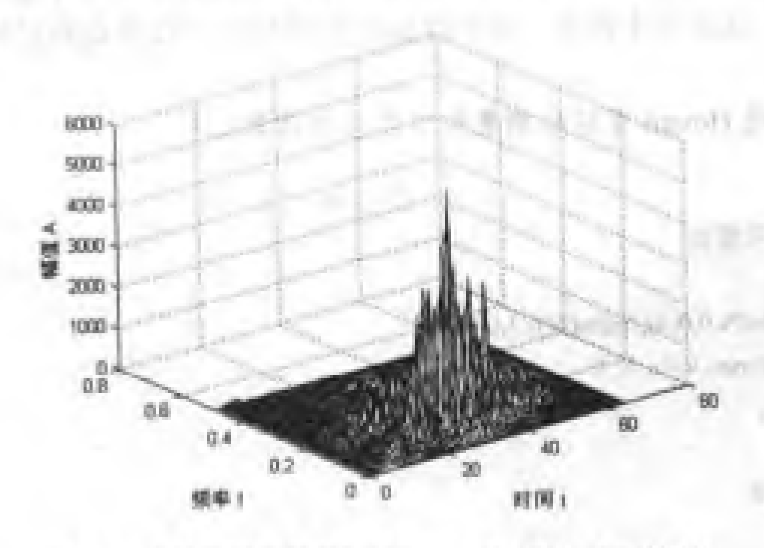


图 7.7　单线性调频信号的 Wigner-ville 分布（三维图）

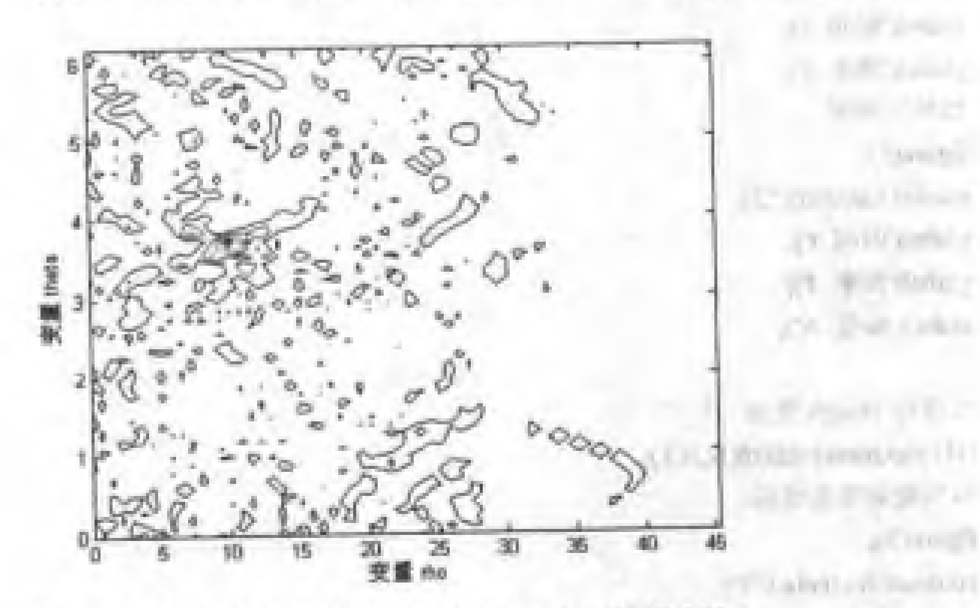


图 7.8　单线性调频信号的 Wigner-Hough 变换（等高线图）

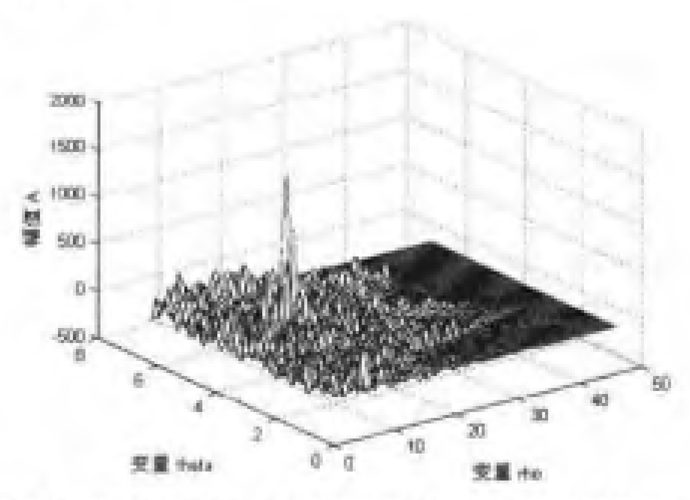


图 7.9　单线性调频信号的 Wigner-Hough 变换（三维图）

由此检验过程很简单：在相应的位置处设置一个阈值，如果峰值大于这个阈值，则说明存在这个信号，反之则不存在。另外峰值的位置提供了线性调频信号的参数估计，图中 $(\rho,\theta)\approx(10,3.8)$。

例程 7-3：利用 Hough 变换检测叠加的两线性调频信号。

f7_3.m

```
%两线性调频信号叠加
N=64;
sig=sigmerge(fmlin(N,0,0.4),fmlin(N,0.3,0.5),1);
%计算信号的 Wibner-Ville 分布
[tfr,t,f]=tfrwv(sig);
%绘图
%%绘制等高线图
figure(1);
contour(t,f,abs(tfr).^2);
xlabel('时间 t');
ylabel('频率 f');
%%三维图
figure(2);
mesh(t,f,abs(tfr).^2);
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');

%进行 Hough 变换
[HT,rho,theta]=htl(tfr,N,N,1);
%%绘制等高线图
figure(3);
contour(rho,theta,HT');
xlabel('rho');
```

```
ylabel('theta');
%%三维图
figure(4);
mesh(rho,theta,HT');
xlabel('rho');
ylabel('theta');
zlabel('幅值 A');
```

两线性调频信号的 Wigner-Ville 分布的等高线图如图 7.10 所示、三维视图如图 7.11 所示。

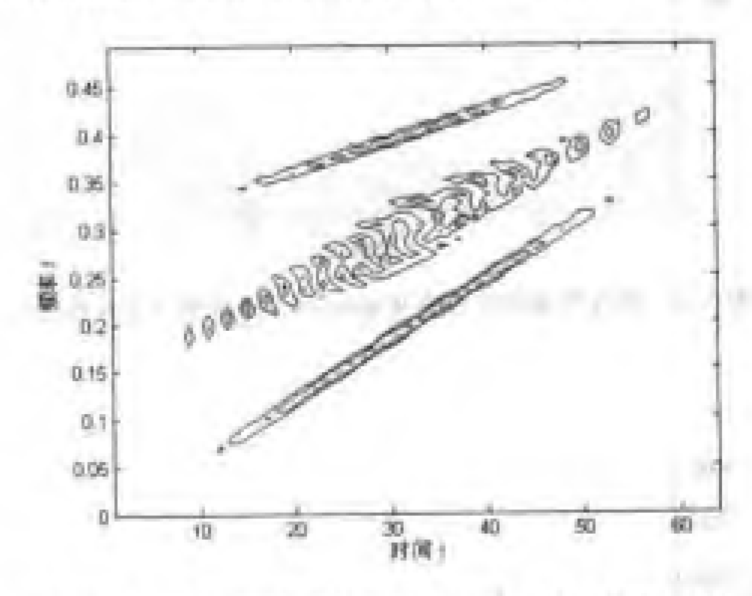


图 7.10　两线性调频信号的 Wigner-ville 分布（等高线图）

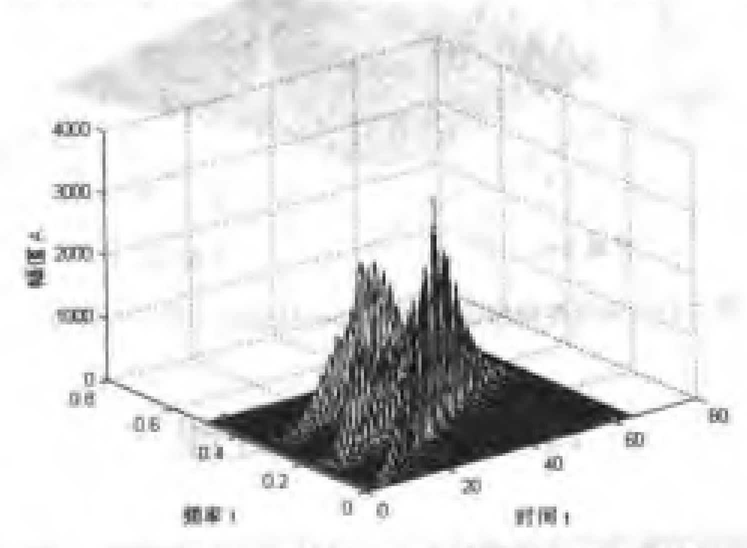


图 7.11　两线性调频信号的 Wigner-ville 分布（三维图）

可以明显看出存在交叉项的问题。接着计算其 Wigner-Hough 变换，如图 7.12（等高线图）和图 7.13（三维图）所示，可以看出在（ρ,θ）平面上存在两个明显分开的峰值，它分别表征了其中的一个线性调频信号，而且根据每个峰值的位置坐标可以估计不同的线性调频信号的参数。由此可见，Wigner-Hough 变换减小了交叉项的干扰，这是因为交叉项本身具有振荡结

构，因此在 Hough 变换中进行积分处理时会降低其影响。

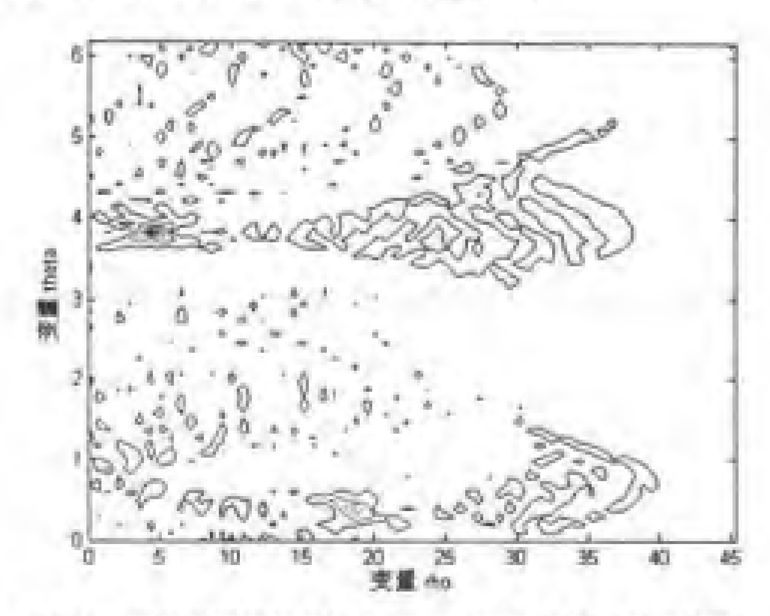


图 7.12　两线性调频信号的 Wigner-Hough 变换（等高线图）

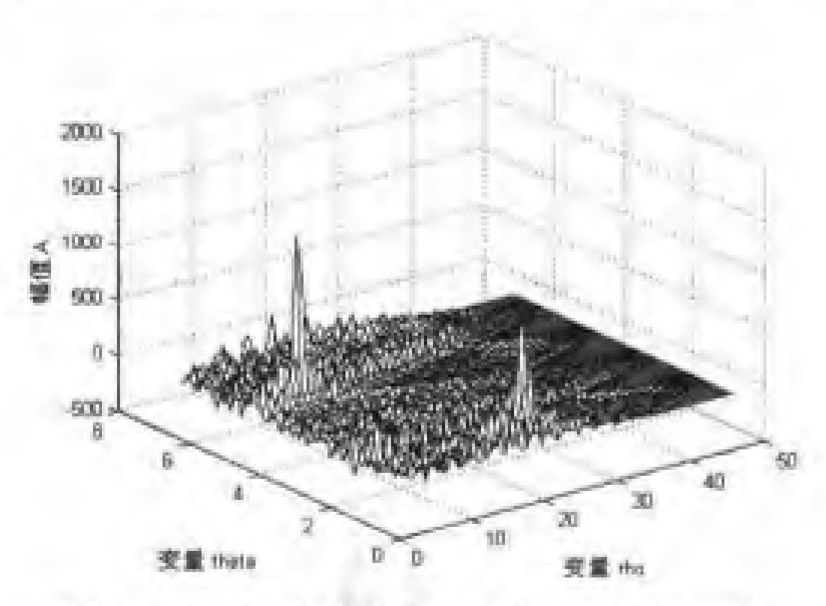


图 7.13　两线性调频信号的 Wigner-Hough 变换（三维图）

7.3　语音信号处理

语音信号处理是研究用数字信号处理技术对语音信号进行处理的一门学科，处理的目的是要得到一些语音参数以便高速地传输或储存，或者通过处理的某种算法以达到某种用途的要求，比如人工合成语音、辨识出讲话者以及识别出讲话的内容等。语音信息处理大致上可以分为三类：（1）人与人之间的语音通信，包括语音压缩与编码、语音增强等；（2）第一类人机语音通信问题，也就是语音的人工合成研究；（3）第二类人机语音通信问题，也就是语音的人工识别和理解的研究。

由于语音信号是随时间变化的，通常认为，语音是一个受准周期脉冲或随机噪声源激励

的线性系统的输出，输出频谱是声道系统频率响应与激励源频谱的乘积，因此标准的傅立叶变换不能直接用于语音信号。由于语音信号可以认为在短时间内近似不变，因而可以采用短时傅立叶变换方法。

7.3.1 短时能量

语音信号的短时能量通常定义为

$$E_n=\sum_{k=-\infty}^{\infty}[x(k)g(n-k)]^2$$

这个表达式还可以写成

$$E_n=\sum_{k=-\infty}^{\infty}x^2(k)h(n-k)$$

其中 $h(n)=g^2(n)$。由此可以看出，语音信号的短时能量就是信号 $x^2(n)$经过冲击响应为$h(n)$数字滤波器滤波后的结果。

语音的短时能量表示取决于窗函数 $h(n)$的选择，常用的有两种窗：

一种是矩形窗

$$h_R(n)=\begin{cases}1,(0\le n\le N-1)\\0,(其\ 他)\end{cases}$$

另一种是 hamming 窗

$$h_H(n)=\begin{cases}0.54-0.46\cos[2\pi n/(N-1)],(0\le n\le N-1)\\0,(其\ 他)\end{cases}$$

研究表明，语音信号的清音段的能量值明显小于浊音段的能量，因此根据短时能量函数可以大致区分浊音和清音。另外在高信噪比的情况下，还可以利用短时能量函数来区分有声和无声。

例程 7-4：利用矩形窗计算语音信号的短时能量。

f7_4.m

```
%读入语音文件
fin=fopen('DR4_MLJH0_SX334.ADC','r');
x=fread(fin,'short');
fclose(fin);
%时域波形
figure(1);
plot(x,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');

%计算短时能量
N=length(x);
%%短时窗长
L=200;
for i=1:N-L
    e(i)=energy(x(i:i+L));
```

```
end
figure(2);

%%能量图
plot(e,'LineWidth',2);
xlabel('时间 t');
ylabel('能量 E');
%energy.m
function [e] = energy(s);
e=sum(s.*s)/max(size(s));
```

读入的语音信号的时域波形如图 7.14 所示，加长度为 200 的矩形窗后计算其短时能量函数如图 7.15 所示，从这个图上能明显区分清音段和浊音段。

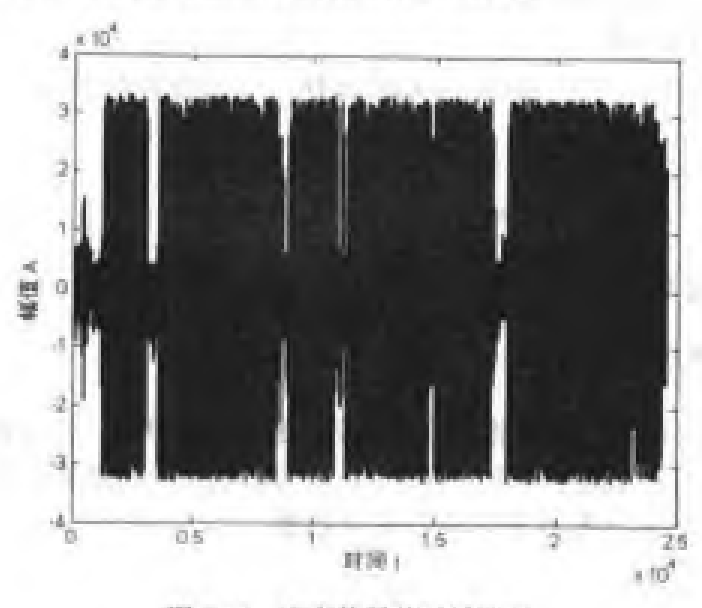


图 7.14　语音信号的时域波形

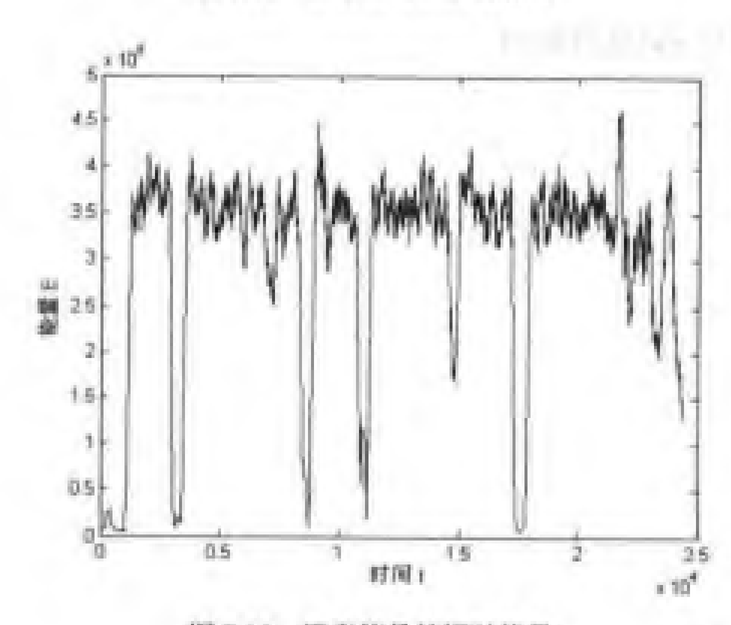


图 7.15　语音信号的短时能量

7.3.2 端点检测

端点检测的目的是从包含语音的一段信号中确定出语音的起点和结束点，这是语音信号分析中的一个很根本的问题。有效的端点检测不仅使处理时间减到最小（帧数最小），而且能排除无声段的干扰，从而使处理质量得到保证，常用的方法是利用短时能量函数。

例程 7-5：利用短时能量检测语音信号的端点。

f7_5.m

```
%读入语音文件
fin=fopen('DR4_MLJH0_SX334.ADC','r');
x=fread(fin,'short');
fclose(fin);

%窗长
l=300;
step=100;
e_x=frame(x,'energy');
figure(1);
plot(e_x,'LineWidth',2);
xlabel('时间 t');
ylabel('帧输出');

s=sort(e_x);
min_e=s(ceil(5*step/64));
max_e=s(length(s)-ceil(5*step/64));

%设置阈值
threshold=min_e+0.2*(max_e-min_e);
a=e_x>threshold;
%门限检验
if ((length(a) < 15*64/step) | (max_e/min_e < 1.3)),
   disp('No speech found');
else
   figure(2);
   plot(a,'LineWidth',2);
   xlabel('时间 t');
   ylabel('端点');
   set(gca,'ylim',[0 1.5]);
end

%frame.m
```

```
function [yy] = frame(x,func,SAMP_FREQ,l,step)
[m,n]=size(x);
if m>n
    n=m;
else
    n=n;
    x=x';
end
if nargin < 3, SAMP_FREQ=8000; end;
if nargin < 4, l=SAMP_FREQ/40; end;
if nargin < 5, step=l/2; end;

%帧的数目
num_frames=ceil(n/step);
%补零
x(n+1:n+2*l)=zeros(2*l,1);
i=[0:step:num_frames*step]';
j=i*ones(1,l);
i=j+ones(num_frames+1,1)*[1:l];
y=reshape(x(i),num_frames+1,l)';
y=(hanning(l)*ones(1,num_frames+1)).*y;
for i=1:num_frames
    cmd=sprintf('yy(:,i)=%s(y(:,i));',func);
    eval(cmd);
end
```

每帧内语音信号的波形如图 7.16 所示，检测的端点如图 7.17 所示。

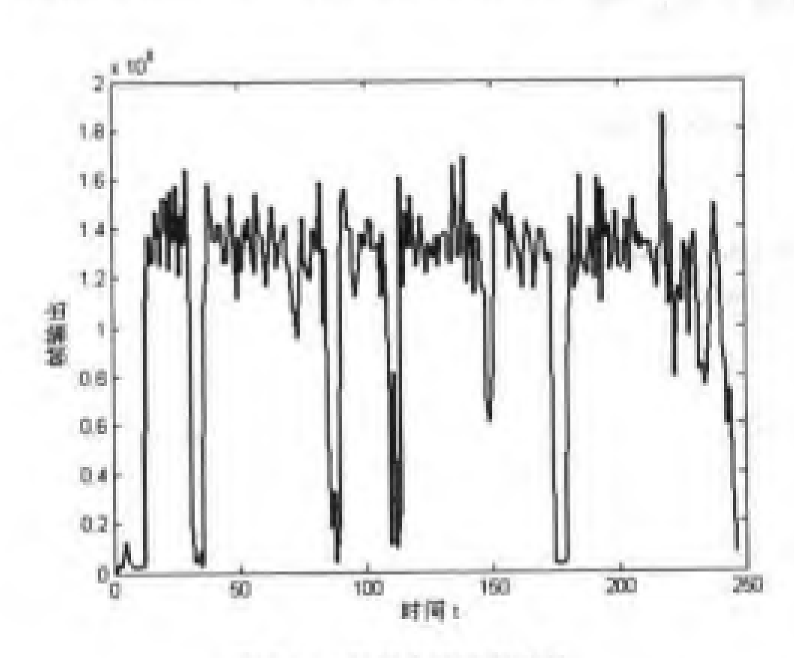


图 7.16　每帧内的语音信号

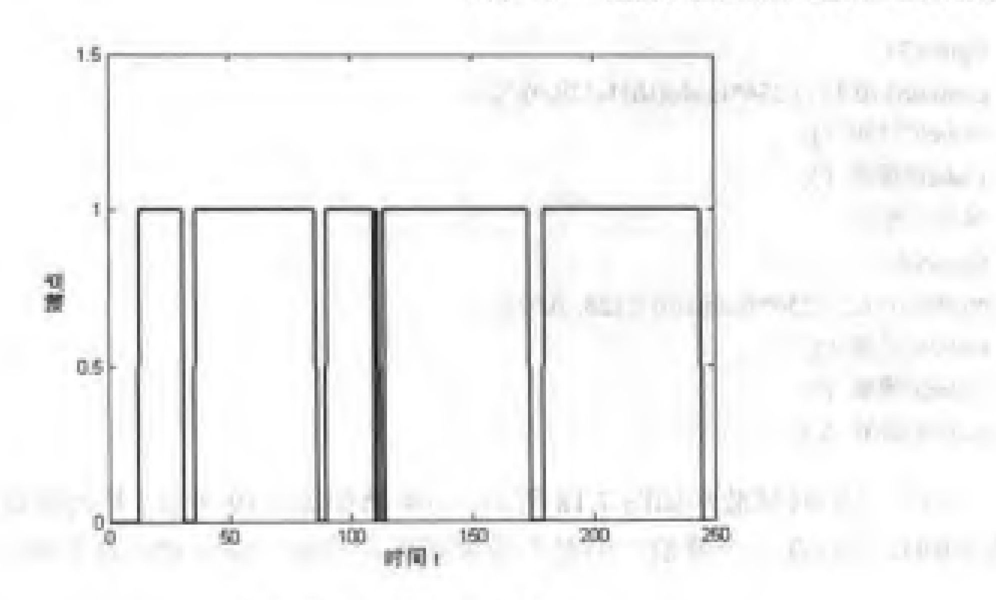


图 7.17　语音信号的端点

7.3.3　语音识别

例程 7-6：分析单词 “GABOR” 的语音信号，样本长度为 338，采样频率为 1kHz。

f7_6.m

```
%读入语音信号
load gabor;
t=0:337;
%采样频率
fs=1000;
%绘制时域波形
figure(1);
plot(t,gabor,'LineWidth',2);
xlabel('时间 t');
ylabel('幅值 A');
%计算功率谱
nfft=10240;
dsp=abs(fft(gabor,nfft));
%绘制功率谱图
figure(2)
plot((0:nfft/2-1)/nfft*fs,dsp(1:nfft/2),'LineWidth',2);
xlabel('频率 f');
ylabel('功率谱 PSD');
%计算短时傅立叶变换
%%窗函数
h=window(85,'hamming');
[tfr,t,f]=tfrstft(gabor,1:338,256,h);
%%绘制等高线图
```

```
figure(3)
contour(t,(0:127)/256*fs,abs(tfr(1:128,:)).^2);
xlabel('时间 t');
ylabel('频率 f');
%%三维图
figure(4)
mesh(t,(0:127)/256*fs,abs(tfr(1:128,:)).^2);
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');
```

语音信号的时域波形如图 7.18 所示，功率谱如图 7.19 所示。从功率谱图中可以看出在频率为 140Hz 处存在一个峰值，但是不能说明单词的哪一部分对应这个峰值。

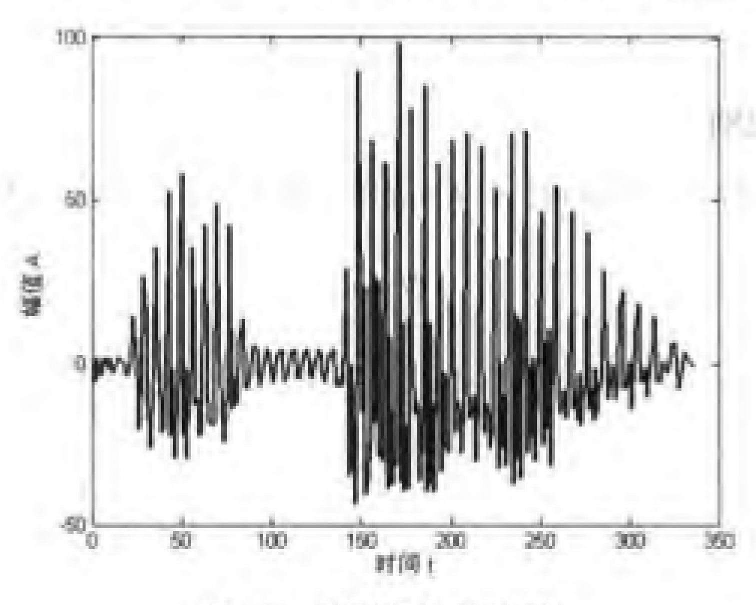


图 7.18 语音信号的时域波形

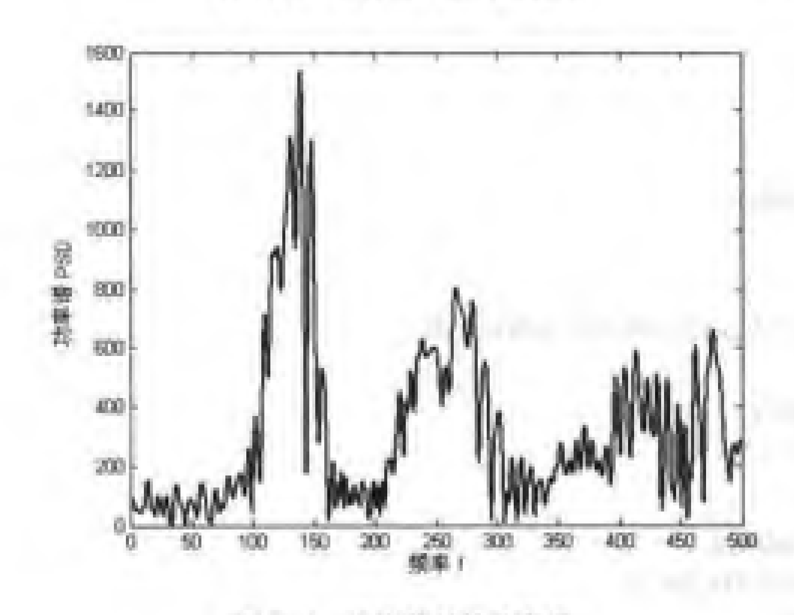


图 7.19 语音信号的功率谱

为此我们计算了语音信号的短时傅立叶变换（使用了一个长度为 85 的 hamming 窗），其等高线图如图 7.20 所示，三维图如图 7.21 所示。从等高线图上可以发现一些有意义的特征：时频面的第一个特征，它位于 30ms~60ms 之间，以 150Hz 为中心，对应于单词的第一个音节“GA”；时频面的第二个特征，它位于 150ms~250ms 之间，对应于单词的最后一个音节“BOR”，它的平均频率随时间从 140Hz 减小到 110Hz，另外，时频面还有这两个频率的谐波分量，只是幅值变小了。

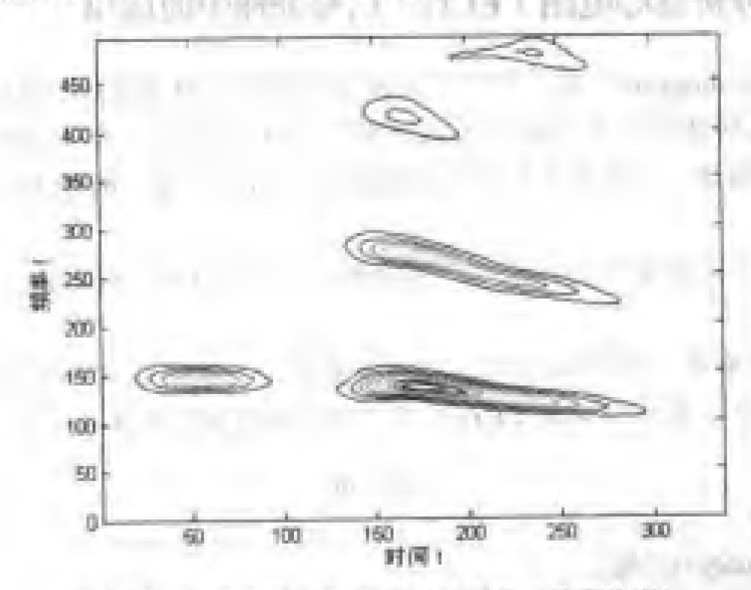


图 7.20　语音信号的短时傅立叶变换（等高线图）

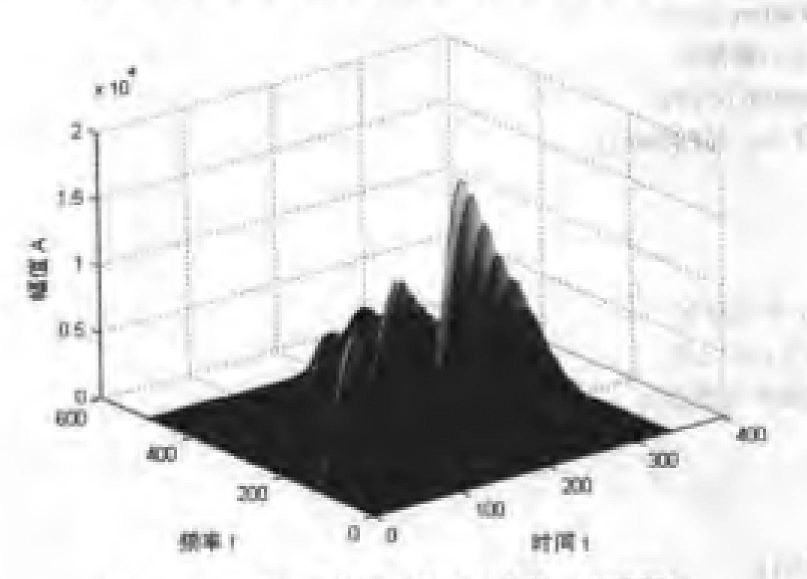


图 7.21　语音信号的短时傅立叶变换（三维图）

7.4　医学信号处理

生物医学信号可以定义为源于一个生物系统的一类信号，像心音、心理和神经系统活动等。这些信号通常含有与生物系统生理和结构状态相关的信息，因此，它们对这些系统状态的研究和诊断具有价值。然而，这些信号是否真正具有使用价值则完全依赖于对其隐含信息的提取能

力，在此领域，生物医学信号处理扮演着一个重要的角色。传统上，医师通过对很多类型波形的观察和一些相关实验来对生物医学信号进行解释，或者假设信号是平稳的或分段平稳的前提下，采用经典的时域及频域方法对信号进行分析。但是在实际中，上述平稳假设往往并不成立，许多医学信号是非平稳的，比如生物声信号，这就需要利用时频分析方法来进行处理。

7.4.1 时频分析在心电图（ECG）信号处理中的应用

ECG（Electrocardiogram）是一种用于诊断心肌状况的无创伤性方法，它由一系列连续的PQRST 波组成，这些波都代表着每个心动周期中的不同状态。为了能够对 ECG 信号的非平稳行为进行精确地描述，传统的 FT 显然不能满足实际的需要，时频表示方法用于 ECG 的分析已经被广泛采用。

实验分别采集了正常窦性心律信号、充血性心脏的心律信号以及心室纤维颤动的心律信号来进行对比分析。

例程 7-7：瞬时频率是时频分析中的一个重要概念，它可以用于估计 ECG 信号的频率变化，这可以用于区分正常和异常的 ECG 信号以及检测 QRS 波群和心律失常。

f7_7.m

```
%正常人心律信号
%fid=fopen('normalecg.txt','r');
%充血性心脏心律信号
fid=fopen('heartfailure.txt','r');
%心室纤维颤动心律信号
%fid=fopen('atrialfib.txt','r');
x=fscanf(fid,'%f %f %f',[300 3]);
fclose(fid);

%采样频率
fs=1/0.008;%  正常信号下
%fs=1/0.004;%充血性心脏
%fs=1/0.004;%心室纤维颤动

N=length(x);
for i=0:N/3-1
    t(i+1)=x(3*i+1);
    sig1(i+1)=x(3*i+2);
    sig2(i+1)=x(3*i+3);
end

%时域波形
figure(1);
subplot(2,2,1);
plot(t,sig1,'LineWidth',2);
xlabel('时间 t/s');
```

```
ylabel('电压  V/mV');

%估计瞬时频率
sig=hilbert(sig1);
sig=sig-mean(sig);
[ifr,tt]=instfreq(sig',2:length(sig)-1);

%显示结果
subplot(2,2,2);
plot(tt/fs,(1-ifr)*fs/2, 'LineWidth',2);
xlabel('时间  t/s');
ylabel('瞬时频率  f/Hz');
```

图 7.22～图 7.24 分别是 3 种不同状态下的 ECG 信号的时域波形及其瞬时频率估计结果，比较可见，正常时和异常时的 ECG 信号波形差别较大，表现在 P、QRS 和 T 波的位置及其形状上。

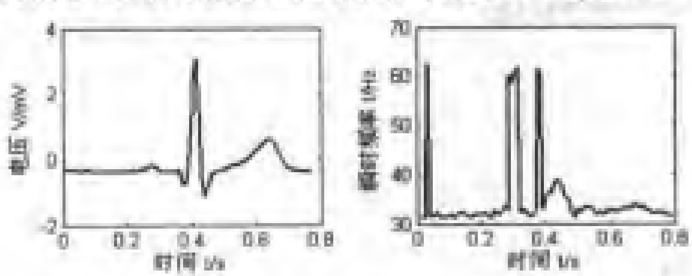


图 7.22　正常 ECG 信号的波形及其瞬时频率

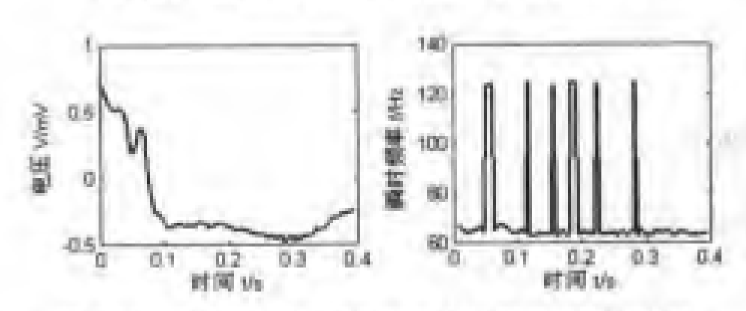


图 7.23　充血性心脏 ECG 信号的波形及其瞬时频率

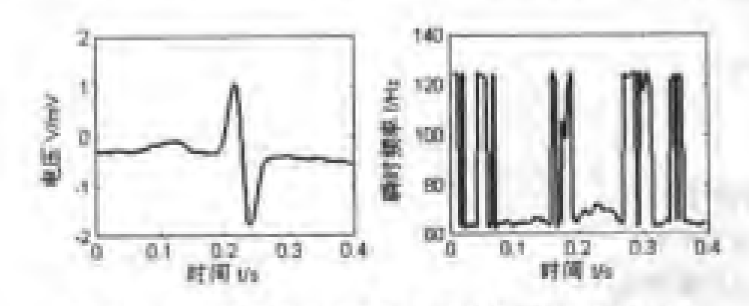


图 7.24　心室纤维颤动 ECG 信号的波形及其瞬时频率

另外 3 种状态下的瞬时频率明显不同，正常的 ECG 信号的频率变化主要发生在 QRS 波的附近，而异常的 ECG 信号的频率变化位置有多处，且无规则。

在 ECG 的解释中，P 波的检测是一个既重要又困难的工作。P 波反映左、右两心房激动

的过程，它能够提供重要的诊断信息。检测 P 波的困难在于它与大幅度的 QRS 波比较临近且是非平稳的，在时频分布中，P 波表现为振荡式样，它的特征主要由两个不同的成分确定。

例程 7-8：利用互 Wigner-Ville 分布分析心电图信号。

f7_8.m

```
clear;
%正常人心律信号
fid=fopen('normalecg.txt','r');
%x=fscanf(fid,' %f %f %f',[300 3]);
%充血性心脏心律信号
%fid=fopen('heartfailure.txt','r');
x=fscanf(fid,' %f %f %f',[300 3]);
%心室纤维颤动心律信号
%fid=fopen('atrialfib.txt','r');
%x=fscanf(fid,' %f %f %f',[300 3]);
fclose(fid);

N=length(x);
for i=0:N/3-1
    t(i+1)=x(3*i+1);
    sig1(i+1)=x(3*i+2);
    sig2(i+1)=x(3*i+3);
end

subplot(2,1,1);
plot(t,sig1,'LineWidth',2);
xlabel('时间 t/s');
ylabel('电压 V/mV');

subplot(2,1,2);
plot(t,sig2,'LineWidth',2);
xlabel('时间 t/s');
ylabel('电压 V/mV');

%采样频率
fs=1/0.008;% 正常信号下
%fs=1/0.004;%充血性心脏
%fs=1/0.004;%心室纤维颤动

x1=sig1;
x2=sig2;

%计算互 Wigner-Ville 分布
```

```
[tfr,t,f]=tfrwv([x1 x2]',1:length(x1),256);

%显示结果
figure(2)
contour(t/fs,f(1:128)*fs,tfr(1:128,:));
xlabel('时间 t/s');
ylabel('频率 f/Hz');
```

实验中采集了两路正常的 ECG 信号，其波形如图 7.25 所示。由图可以看出，ECG 信号中的 QRS 和 T 波比较明显，但是 P 波难以看出。

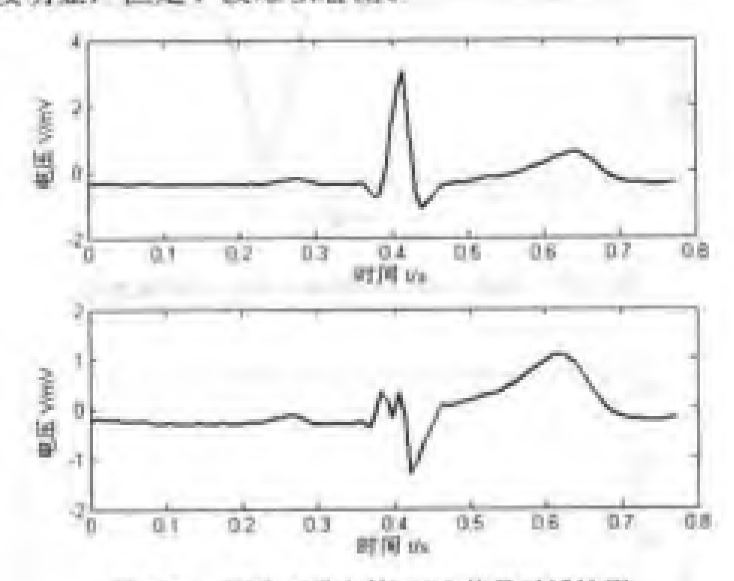


图 7.25　两路正常人的 ECG 信号时域波形

为此，计算了两路 ECG 信号的互 Wigner-Ville 分布，结果如图 7.26 所示。由图可以看出在 0.2s～0.4s 之间有振荡频率的存在，这预示了 P 波的位置。

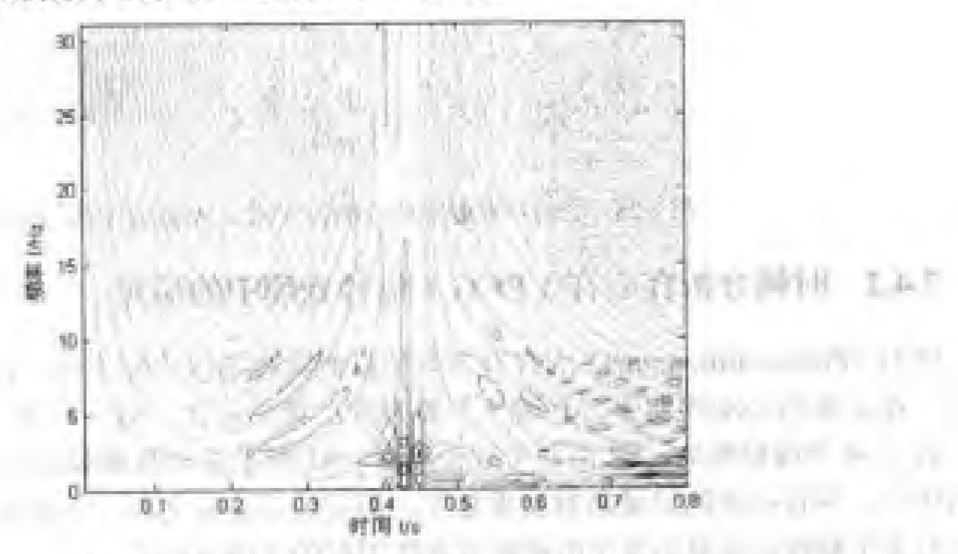


图 7.26　正常人心律信号的互 Wigner-Ville 分布

进一步为了比较异常 ECG 信号与正常 ECG 信号的时频分布的区别，接着计算了心室纤维颤示，可见两者之间存在明显的区别，其一是峰值的位置出现了变化，其二是频率范围发生了变化。

颤动的心律信号（两路信号的时域波形如图 7.27 所示）的互 Wigner-Ville 分布，如图 7.28 所示。

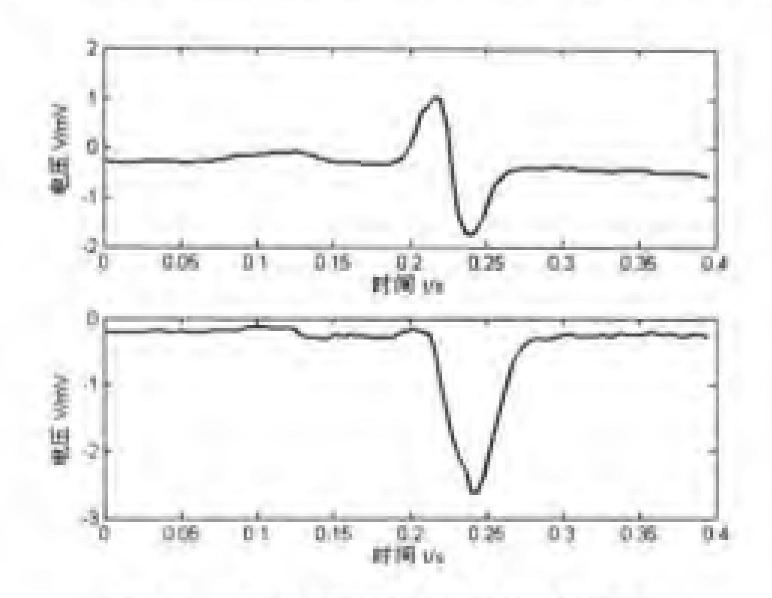


图 7.27　两路心室纤维颤动心律信号时域波形

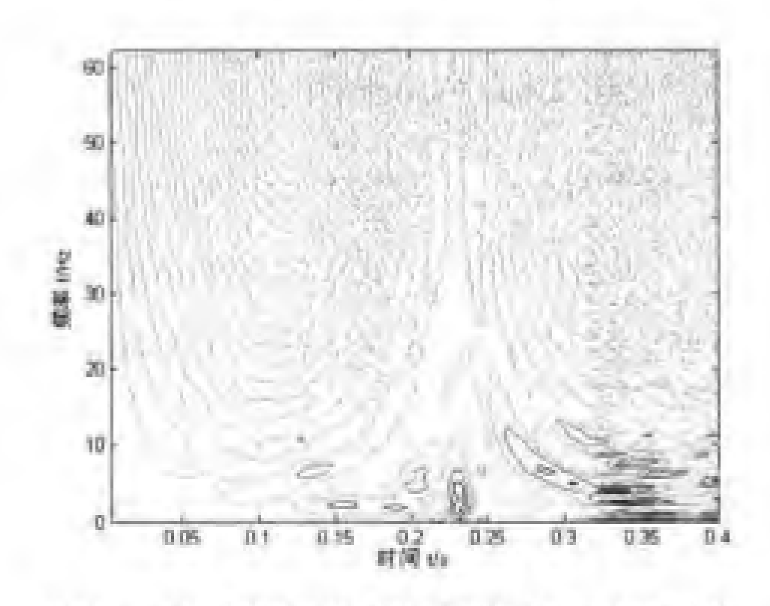


图 7.28　心室纤维颤动的心律信号的互 Wigner-Ville 分布

7.4.2　时频分析在心音（PCG）信号处理中的应用

PCG（Phonocardiograms）因含有预知信息而具有临床诊断价值，它能够在胸部表面上测量到。在正常的心动周期中，有两个主要声音：第一心音（S1）和第二心音（S2）。普遍认为，S1 由 4 个成份组成，S2 由两个成份组成。S1 源于主动脉瓣膜的闭合，S2 源于肺动脉瓣膜的闭合。尽管心音的记录具有可重复性，但它们不能定义为一个简单的周期信号。心音信号具有非平稳性而且是由复杂的机械声事件引起的多成份信号。

传统的谱分析方法反映的是信号的静态频谱特征，而反映心脏及心血管系统生理、病理特征的心音信号通常表现为非平稳时变特征，因此采用经典谱分析方法难以揭示心音信号的动态变化过程，心音信号的时频分析对于心脏系统方面的基础研究和临床诊断有着十分重要

的意义，本小节将利用二次型时频分布来分析心音信号。

例程 7-9：正常和异常时的心音信号波形。

f7_9.m

```
%样本点数
N=5000;
[normal,Fs,bits] = wavread('normal',N);
[mitralstenosis,Fs,bits] = wavread('mitralstenosis',N);
[aorticinsufficiency,Fs,bits] = wavread('aorticinsufficiency',N);
[pulmonarystenosis,Fs,bits] = wavread('pulmonarystenosis',N);

figure(1);
%正常时心音波形
subplot(2,1,1);
plot(normal);
xlabel('样本序号 n/个');
ylabel('幅值 A/V');
title('正常时心音波形');
axis ([0 N -0.1 0.1]);

figure(2);
%大动脉狭窄时心音波形
subplot(3,1,1);
plot(aorticinsufficiency);
% xlabel('样本序号 n/个');
ylabel('幅值 A/V');
title('大动脉狭窄时心音波形');
axis ([0 N -0.2 0.2]);
%二尖瓣狭窄时心音波形
subplot(3,1,2);
plot(mitralstenosis);
% xlabel('样本序号 n/个');
ylabel('幅值 A/V');
title('二尖瓣狭窄时心音波形');
axis ([0 N -0.1 0.1]);

%肺动脉瓣狭窄时心音波形
subplot(3,1,3);
plot(pulmonarystenosis);
xlabel('样本序号 n/个');
ylabel('幅值 A/V');
title('肺动脉瓣狭窄时心音波形');
axis ([0 N -0.2 0.2]);
```

正常的心音信号波形如图7.29所示，由图7.29明显能看出第一心音和第二心音的波峰值。而异常时的心音信号如图 7.30 所示，由图 7.30 可以看出第一心音和第二心音的峰值位置发生了改变，而且它们的形状还比较模糊，甚至难以区分。

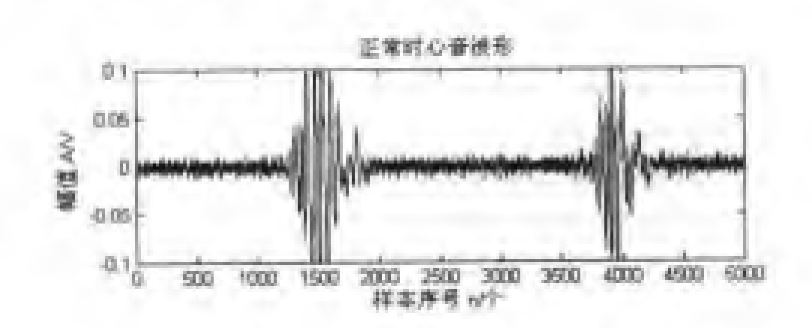


图 7.29　正常的心音信号波形

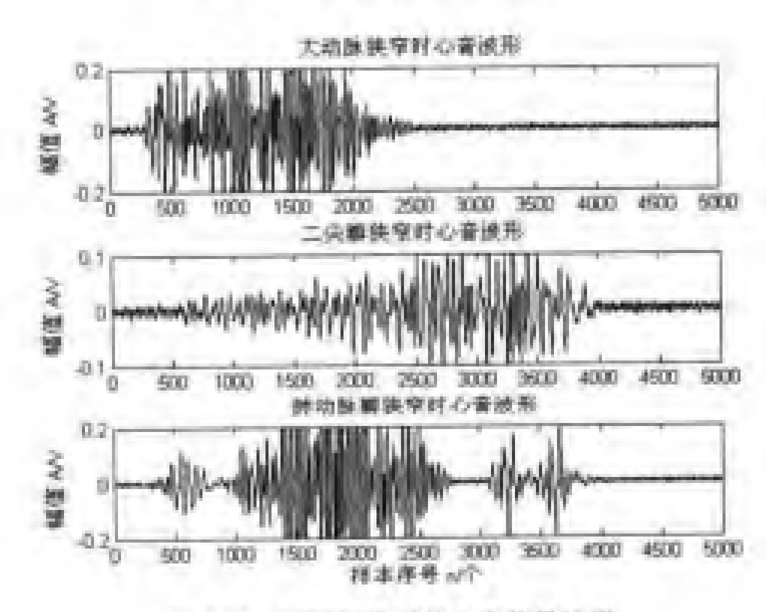


图 7.30　不同异常时的心音信号波形

例程 7-10：利用 Choi-Willams 分布分析心音信号。

f7_10.m

```
%样本点数
N=4096;
%正常时心音信号
[x,fxs,bits] = wavread('normal',N);
%肺动脉瓣狭窄时心音信号
[y,fys,bits] = wavread('pulmonarystenosis',N);

%设置参数
%时域窗函数
g=window(9,'Kaiser');
%频域窗函数
h=window(27,'Kaiser');
```

```
t=1:N;

%计算 Choi-Willams 分布
%%正常状态
x=hilbert(x);
x=x-mean(x);
[tfr,t,f]=tfrcw(x,t,128,g,h,10);
%%显示:三维图
figure(1)
mesh(t,f*fxs/2,abs(tfr));
xlabel('样本序号  n/个');
ylabel('频率  f/Hz');
zlabel('幅值  A/V');
%%等高线图
figure(2)
contour(t,f*fxs/2,abs(tfr));
xlabel('样本序号  n/个');
ylabel('频率  f/Hz');

%%异常状态
y=hilbert(y);
y=y-mean(y);
[tfr,t,f]=tfrcw(y,t,128,g,h,10);

%%显示:三维图
figure(3)
mesh(t,f*fys/2,abs(tfr));
xlabel('样本序号  n/个');
ylabel('频率  f/Hz');
zlabel('幅值  A/V');

%%等高线图
figure(4)
contour(t,f*fys/2,abs(tfr));
xlabel('样本序号  n/个');
ylabel('频率  f/Hz');
```

正常心音信号的 Choi-Willams 分布分别如图 7.31、图 7.32 所示，从图中能明显看出第一心音峰 S1 和第二心音峰 S2。由于这两个峰相隔较远，因此位于 S1 和 S2 之间的交叉项几乎被完全抑制了。进一步观察频率可见，第一心音峰 S1 的主要频率范围为 30~120Hz，而第二心音峰 S2 的主要频率范围为 40～150Hz。

异常心音信号的 Choi-Willams 分布分别如图 7.33、图 7.34 所示，从图中能明显看出第一心音峰 S1 和第二心音峰 S2 的位置与正常时的相比发生了很大变化，而且频率范围也发生了改变。

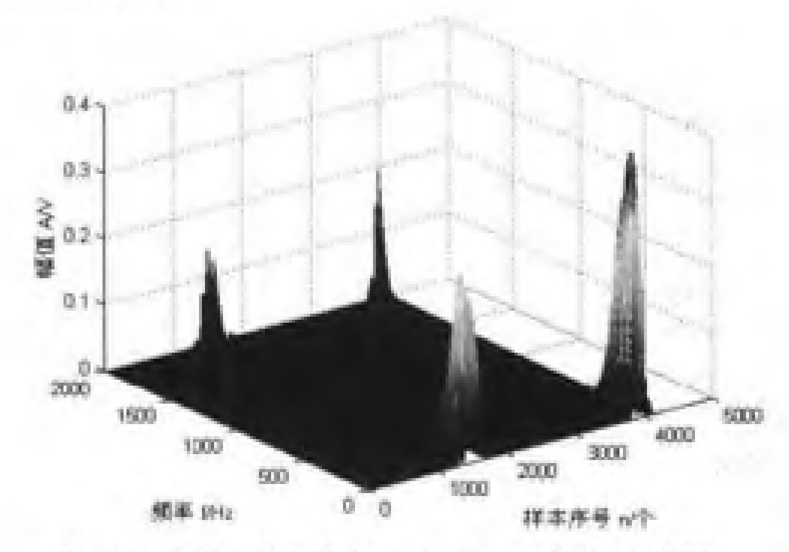


图 7.31　正常心音信号的 Choi-Willams 分布（三维图）

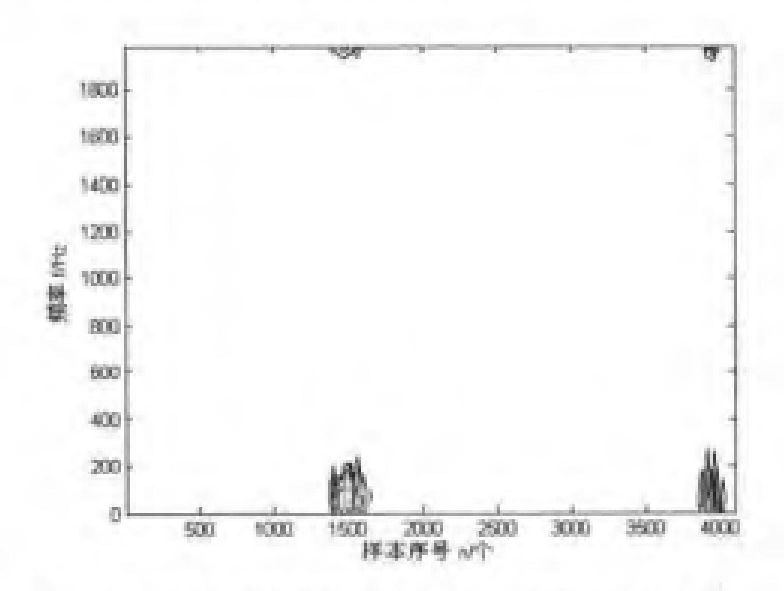


图 7.32　正常心音信号的 Choi-Willams 分布（等高线图）

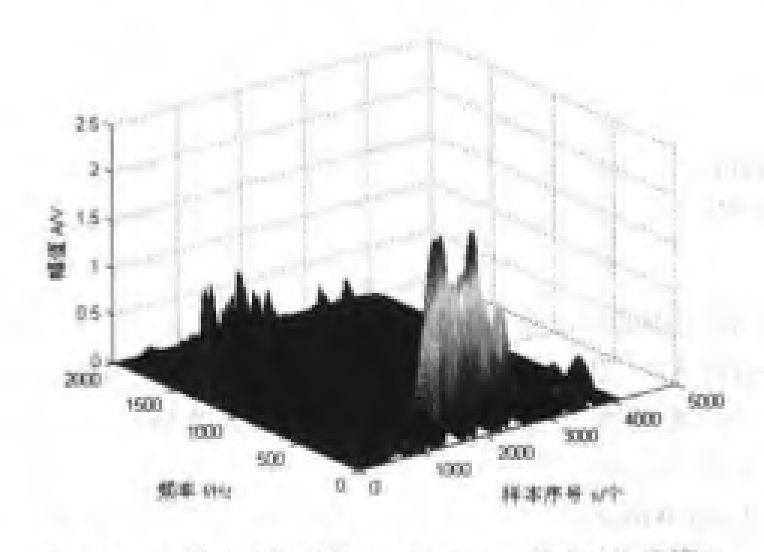


图 7.33　异常心音信号的 Choi-Willams 分布（三维图）

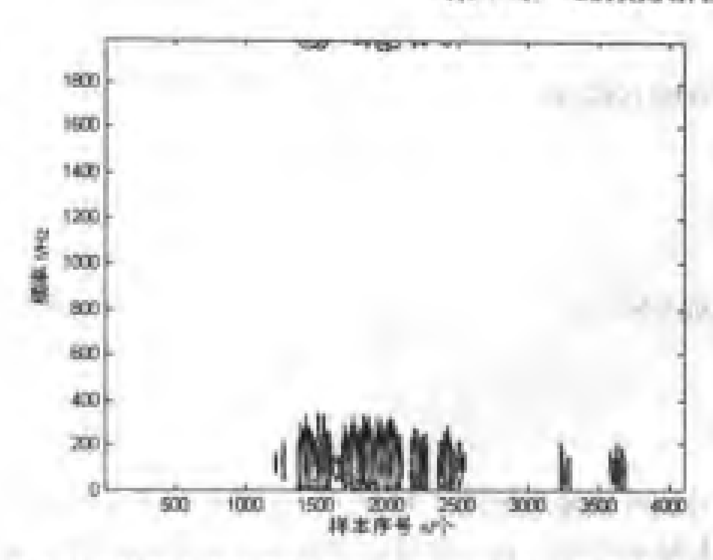


图 7.34　异常心音信号的 Choi-Willams 分布（等高线图）

7.4.3　时频分析在脑电图（EEG）信号处理中的应用

EEG 是一种研究大脑电活动的无创伤方法。EEG 有 3 种类型：非瞬态自发性 EEG、瞬态自发性 EEG 和诱发性 EEG（诱发电位，EP）。傅立叶变换和线性模型方法已广泛地用于分析 EEG 的模式特征和非瞬态自发脑电活动，但是，用仅适合平稳信号的谱分析方法分析瞬态自发性 EEG 和诱发电位这样的非平稳信号是不合适的，需要利用时频分析方法。

脑电信号中往往会出现一些短时瞬态脉冲，这些瞬态脉冲信号有的含有重要病理信息，有的属于干扰信号。无论是哪种类型的瞬态信号，对其进行检测、定位和提取都是脑电信号分析中一个非常重要而且经常遇到的问题。匹配滤波是一种常用传统的方法，但是脑电信号的非平稳性很强，实际中使用这种方法难度大。本小节将使用谱图来提取脑电图信号中的瞬态脉冲。

例程 7-11：利用谱图分析脑电图信号。

f7_11.m

```
%读入 EEG 信号
load eeg.dat;
sig=eeg(2,1:512)';
N=length(sig);
%采样频率
fs=312.5;

%时域波形
figure(1);
plot((1:N)/fs,sig,'LineWidth',2);
xlabel('时间 t/s');
ylabel('电压 A/uV');

%计算谱图
[tfr,t,f]=tfrsp(sig);

%显示
```

```
figure(2);
contour(t/fs,f(1:N/2)*fs,tfr(1:N/2,:));
xlabel('时间 t/s');
ylabel('频率 f/Hz');

figure(3)
mesh(t/fs,f(1:N/2)*fs,tfr(1:N/2,:));
xlabel('时间 t/s');
ylabel('频率 f/Hz');
zlabel('幅值 A/ uV ');
```

采集的脑电图信号的时域波形如图 7.35 所示，由图可以看出在时域波形上存在多处脉冲信号，但是不能确定其频率范围，也不能排除干扰引起的瞬态脉冲。为此计算其谱图分别如图 7.36、图 7.37 所示，可见在 0.3s～0.4s 以及 1.2s～1.4s 之间存在瞬态脉冲信号。

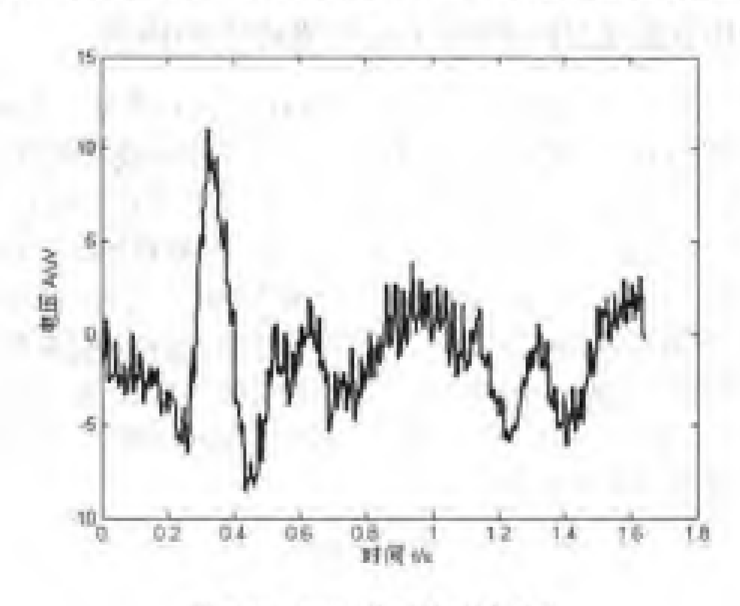


图 7.35　EEG 信号的时域波形

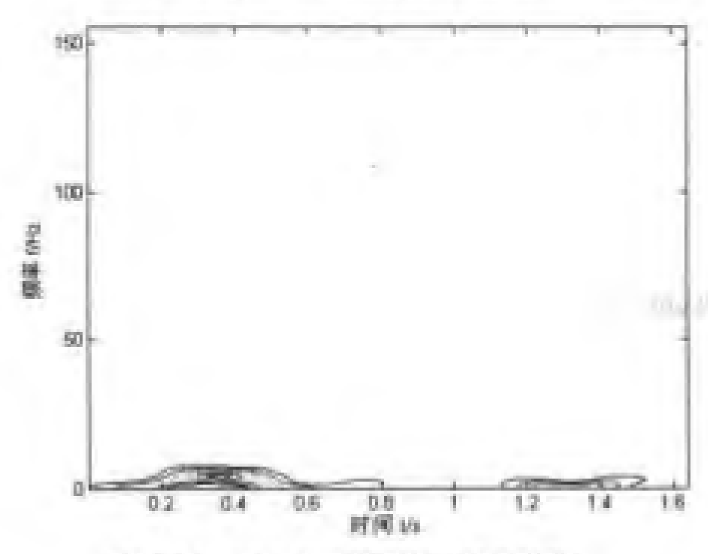


图 7.36　EEG 信号的谱图（等高线图）

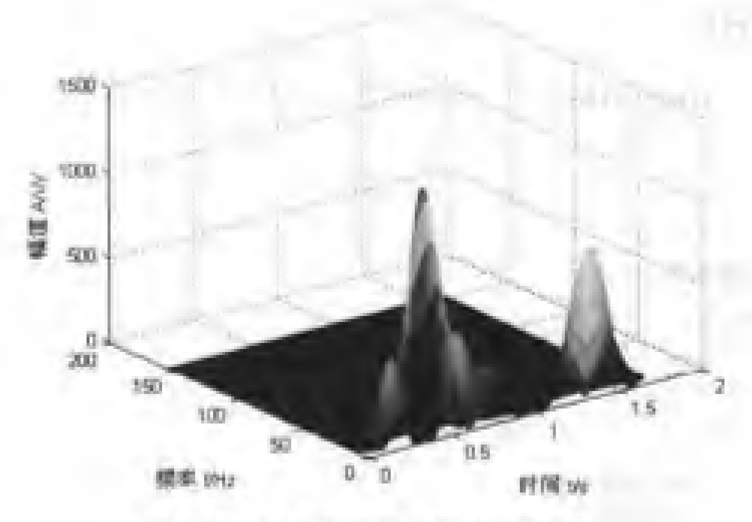


图 7.37　EEG 信号的谱图（三维图）

7.5　合成孔径雷达（SAR）信号处理

7.5.1　基本原理

运动目标成像问题是合成孔径雷达（SAR）信号处理的一个难题，常规 SAR 的距离向高分辨率是通过线性调频信号的脉冲压缩技术获得的，方位向的高分辨率是通过接受数据与一个理论上的静止目标的冲击响应作匹配相关获得的。因此如果成像场景中有以前未知方式运动的目标，常规 SAR 的成像方法就不能正常工作，从而造成运动目标图像的模糊和方位偏移。

SAR 信号的频率是随时间连续变化的，因此它是一个典型的非平稳信号，可以利用时频分析方法提取瞬时频率曲线，而且时频分析方法并不需要知道关于目标的先验知识，它能同时完成检测和参数估计。

SAR 静止目标的后向散射信号可以表示为：

$$S(x)=a(x)\exp\{-jkx^2/R_0\},x\in V_aT$$

式中 T 为合成孔时间，$k=2\pi/\lambda$，λ为工作波长，$a(x)$是天线方位向权函数，V_a是雷达平台速度，R_0是目标与雷达平台之间的距离。

如果目标是运动的，并设（V_r,a_r）是运动目标的径向速度和加速度，（V_c,a_c）是运动目标的方位向的速度和加速度，则散射信号可以表示为：

$$S_m=a[(1-\varepsilon_c^{'})x]\exp(-2jk\varepsilon_r^{'}x)\exp\{-jkx^2[(1-\varepsilon_c^{'})^2-\varepsilon_r^{''}]/R_0\}$$

式中 $\varepsilon_r^{'}=V_r/V_a,\varepsilon_c^{'}=V_c/V_a,\varepsilon_r^{''}=a_rR_0/V_a^2$。

由此可见，运动目标和静止目标的主要差别在于它们的中心频率和调频率不同。SAR 运动目标回波信号是一种复杂的时变信号，联合时频分析方法是解决这一问题的有效工具。常用的 Wigner-Ville 分布能满足要求，但是它也有致命的缺点，由于它是双线性变换，因此用于多点目标和面目标时会出现交叉项干扰，严重降低了它的性能，为此本小节将利用 Hough 变换来抑制交叉项。

7.5.2 例程分析

例程 7-12：利用 Wigner-Hough 变换分析两个目标的 SAR 信号。

f7_12.m

```
clear;
%仿真合成孔径雷达信号
colormap(gray(256))
cj=sqrt(-1);
pi2=2*pi;

c=3e8;                    % 传播速度
fc=200e6;                 % 频率
lambda=c/fc;              % 波长
k=pi2/lambda;             % 波数
Xc=1.e3;                  % 距离目标中心的距离

L=400;                    % 合成孔径大小为 2*L
Y0=100;                   % 目标区域位于[Yc-Y0,Yc+Y0]
Yc=0;                     % 距离目标中心的跨距离

theta_c=atan(Yc/Xc);  %到目标中心的斜视角
Rc=sqrt(Xc^2+Yc^2);   %到目标中心的斜视距离
kus=2*k*sin(theta_c); %Doppler 频率偏移

Xcc=Xc/(cos(theta_c)^2);
DY=(Xcc*lambda)/(4*L);              % 交叉距离分辨率
L_min=max(Y0,L);                    % 补零的孔径

% 压缩信号的 u 域参数和阵列
du=(Xcc*lambda)/(4*(Y0+L));         % 孔径域的采样间隔
duc=(Xcc*lambda)/(4*Y0);            %压缩信号孔径域的采样间隔
mc=2*ceil(L_min/duc);
uc=duc*(-mc/2:mc/2-1);
dkuc=pi2/(mc*duc);                  % ku 域的采样间隔
kuc=dkuc*(-mc/2:mc/2-1);            % kuc 阵列
%
dku=dkuc;                           % ku 域的采样间隔

%合成孔径信号的 u 域参数和阵列
m=2*ceil(pi/(du*dku));              %样本数
du=pi2/(m*dku);
u=du*(-m/2:m/2-1);                  % 合成孔径阵列
```

```
ku=dku*(-m/2:m/2-1);                    % ku 阵列
%
ntarget=2;                   % 目标数
%目标的坐标和反射率
yn(1)=.7*Y0;             fn(1)=0.5;
yn(2)=yn(1)-4*DY;        fn(2)=1;

%测量的回波信号
s=zeros(1,mc);
for i=1:ntarget;
 dis=sqrt(Xc^2+(Yc+yn(i)-uc).^2);
 s=s+fn(i)*exp(-cj*2*k*dis).*(abs(uc) <= L);
end;
figure(1);
plot(uc,real(s));
xlabel('合成孔径 u');
ylabel('幅值 A');
title('孔径信号的实部');
axis([uc(1) uc(mc) 1.1*min(real(s)) 1.1*max(real(s))]);

sig=s(1:50);
%计算 Wigner-Ville 分布
[tfr,t,f]=tfrwv(sig',1:length(sig),256);
%显示 Wigner-Ville 分布等高线
figure(2);
mesh(t,f,tfr);
xlabel('时间 t');
ylabel('频率 f');
zlabel('幅值 A');

%Hough 变换
[WH,rho,theta]=htl(tfr);
figure(3);
mesh(rho,theta,WH');
xlabel('极半径 rho');
ylabel('角度 theta');
zlabel('幅值 A');
```

数值仿真生成的合成孔径回波信号的实部波形如图 7.38 所示，从时域图上不能看出目标的存在；计算其 Wigner-Ville 分布如图 7.39 所示，由于没有噪声的干扰，因此在图上也能看出两个目标的存在，但存在有交叉项。

进一步对 Wigner-Ville 分布进行 Hough 变换，结果如图 7.40 所示，从图上能明显看出两处存在峰值，它对应了两个面目标。

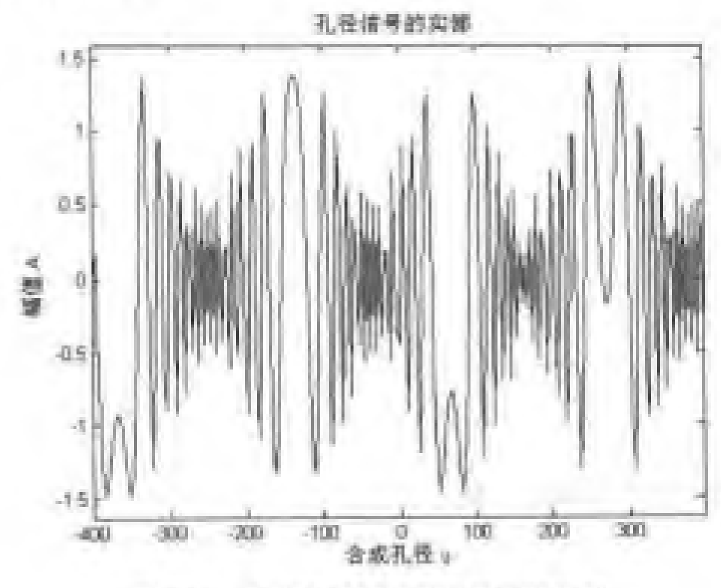


图 7.38　合成孔径回波信号的实部波形

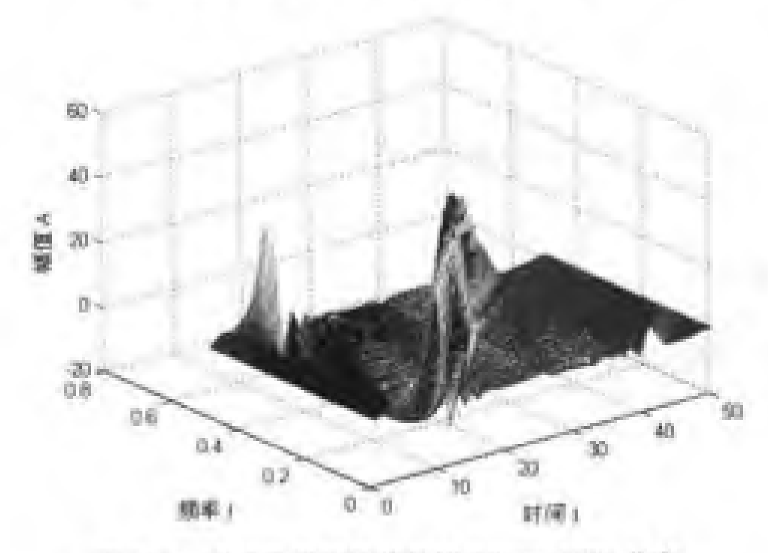


图 7.39　合成孔径回波信号的 Wigner-Ville 分布

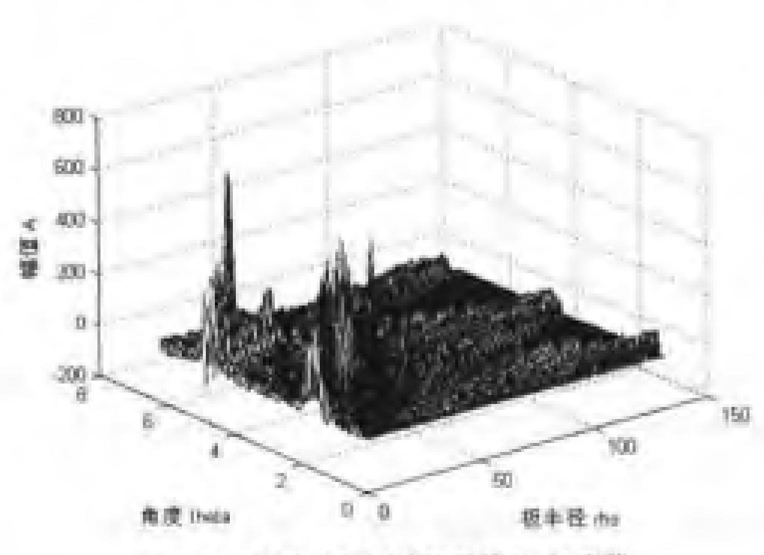


图 7.40　合成孔径回波信号的 W-H 变换

第8章　时频分布在故障诊断中的应用

二次型时频分布可以同时从时域特征和频域特征联合揭示信号的构成本质，通过设计合理的核函数可以使得时频分布具有良好的时频聚集性，能够有效地实现信噪分离，从干扰信号中清晰地分辨出故障特征信号，因此为机械故障诊断提供了一个有利的工具。

本章主要内容包括：

- 时频分布在轴承故障诊断中的应用；
- 时频分布在齿轮故障诊断中的应用。

8.1　引　　言

具体说来，完整的机械设备故障诊断过程应包含以下5个环节：

（1）信号采集：根据机械设备的工作环境和性质，选择并采集能反映轴承工作情况或状态的信号。

（2）特征提取：从采集的信号中以一定的信号分析与处理方法提取出能够反映机械设备状态的有用信息（特征）。

（3）监测（状态识别）：根据征兆，以一定的状态识别方法识别机械设备的状态，既简单判断机械设备工作是否正常或有无故障。

（4）诊断（状态分析）：根据征兆，进一步分析有关状态的情况及其发展趋势，当机械设备有故障时，详细分析故障的类型、性质、部位、产生原因及趋势等。

（5）决策干预：根据机械设备状态及其发展趋势，做出决策，如调整、控制、维修或继续监视等。

振动法是通过安装在轴承座或箱体适当方位的振动传感器监测轴承振动信号。并对此信号进行分析与处理判断轴承工况与故障的。由于振动法具有：（1）适用于各种类型各种工况的轴承；（2）可以有效地诊断出早期微小故障；（3）信号测试与处理简单、直观；（4）诊断结果可靠等优点，所以在实际中得到了极为广泛的应用。

实际上通过安装在轴承座上的传感器采集的振动信号出反映有关轴承本身工作状态的信息外，也包含了大量的机械中其他运动部件和结构的信息，这些信息对于研究轴承本身的工况和故障来说，属于背景噪声。由于实际中背景噪声往往比较大，所以刚刚发生的轻微滚动轴承所特有的信息往往淹没在背景噪声中，很难被发现和提取出来，因此采用什么样的信号处理技术来提高信噪比，突出故障特征信息，抑制背景噪声，从而有效地诊断出轴承故障，成为轴承振动监测与诊断技术的关键所在。

绝大多数机械振动信号都属于非平稳信号，其典型特征时频谱范围很广，成分复杂，而且又往往淹没在噪声和无用的信号之中，所以如何有效地剔除无用信号，刻画故障信号的本

质，就成为机械故障诊断的核心课题。

二次型时频分布为机械故障诊断提供了有利的工具，其基本思想是设计时间和频率的联合函数，并通过核函数的巧妙设计实施有效地过滤，去除噪声，用它来描述信号在不同时间和频率上的能量密度和强度，从时域、频域和幅值域三维空间同时揭示信号成分，可以成功地应用于机械故障诊断。

作为基础零部件的轴承和齿轮广泛应用于各种机械设备中。轴承和齿轮的状态与设备的运行直接相关，因此轴承和齿轮运行状态的识别与诊断一直是人们关注的研究课题，研究轴承及齿轮缺陷和故障的识别技术和方法无疑具有重要意义。本章将利用时频分布对轴承和齿轮的各种故障进行检测，由于直接使用时频分布会存在严重的交叉项，因此本节中的时频分布都进行了加窗处理。

8.2 时频分布在轴承故障诊断中的应用

轴承是旋转机械中的一类重要部件，当滚动轴承出现局部损伤时，在受载运转过程中，轴承的其他零件会周期地撞击损伤点，产生的冲击力激励轴承座及其支撑结构，形成一系列由冲击激励产生的减幅振荡，减幅振荡发生的频率称为故障特征频率，它由轴的转速、轴承几何尺寸及损伤点的位置（外圈、内圈、滚动体）确定，根据故障特征频率可以检测轴承是否出现故障并确定故障的位置，它们的计算公式可参照相关文献。

本节在某汽车变速器上分别完成了轴承正常状态、轴承外环故障和轴承滚动体故障三次实验，振动信号利用振动传感器从某减速器的箱体上采集，采样频率为 10kHz，下面将利用 Choi-Williams 分布对这些数据进行分析，以提取不同故障的特征频率。

8.2.1 计算正常轴承信号的 Choi-Williams 分布

f8_1.m

```
%实验信号
N=1024;
t=1:N;
fid=fopen('normal.dat','r');%正常信号
%fid=fopen('GEARch2_3.dat','r');%故障信号
sig=fread(fid,N,'int16');
fclose(fid);
%归一化
sig=(sig-mean(sig))/std(sig,1);
%采样频率
fs=10000;

%时域波形
figure(1)
plot(t/fs,sig);
xlabel('时间 t/s');
ylabel('幅值 A/V');
%功率谱
```

```
nfft=10240;
y=abs(psd(sig,nfft));
figure(2)
plot((0:nfft/2-1)/nfft*fs,y(1:nfft/2));
xlabel('频率 f/Hz');
ylabel('功率谱 S/V');

%计算 Choi-Williams 分布
sig=hilbert(sig);
[tfr,t,f]=tfrcw(sig);
figure(3)
contour(t/fs,f(1:N)*fs,abs(tfr));
xlabel('时间 t/s');
ylabel('频率 f/Hz');
```

正常状态下采集的振动信号如图 8.1 所示，其功率谱如图 8.2 所示。由功率谱图可以看出，只存在齿轮的啮合频率，这说明正常状态下主要是齿轮的啮合信号。

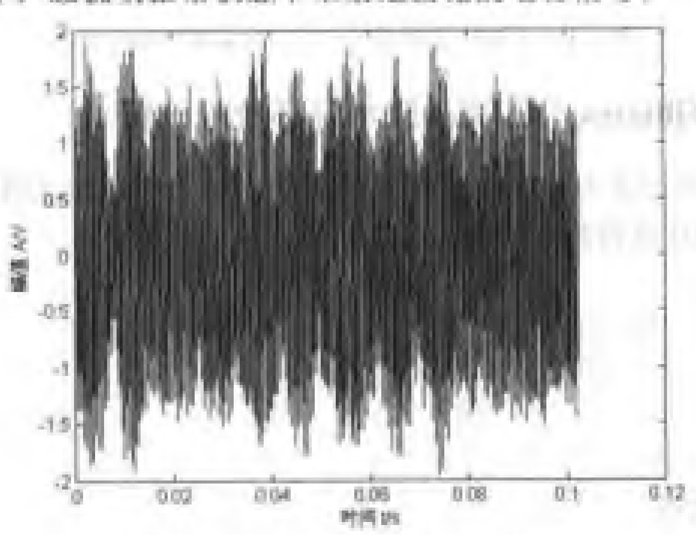


图 8.1 正常状态振动信号波形

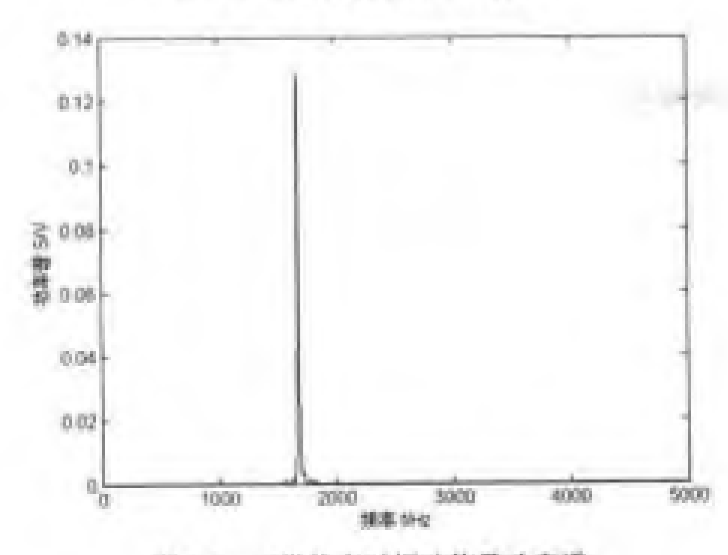


图 8.2 正常状态时振动信号功率谱

计算正常状态下振动信号的 Choi-Williams 分布如图 8.3 所示，同样可以看出图中的两条直线分别对应于齿轮的啮合频率及其倍频。

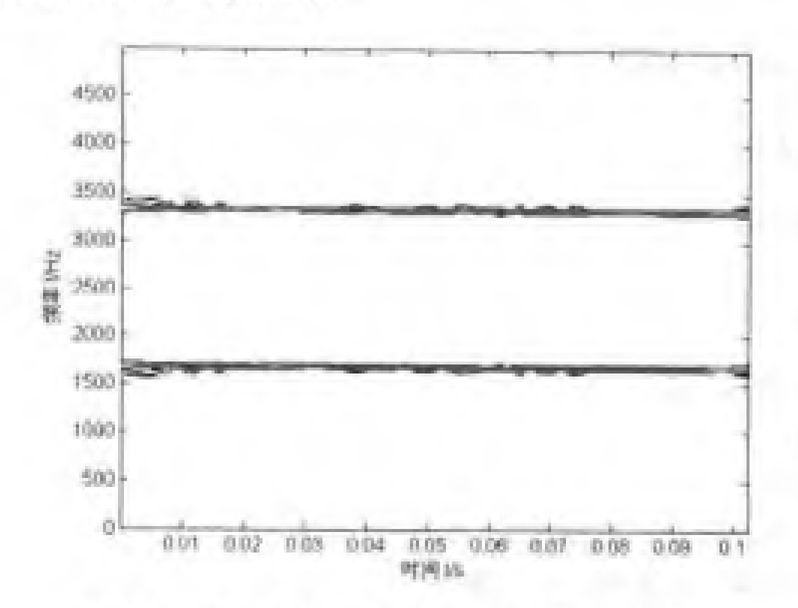
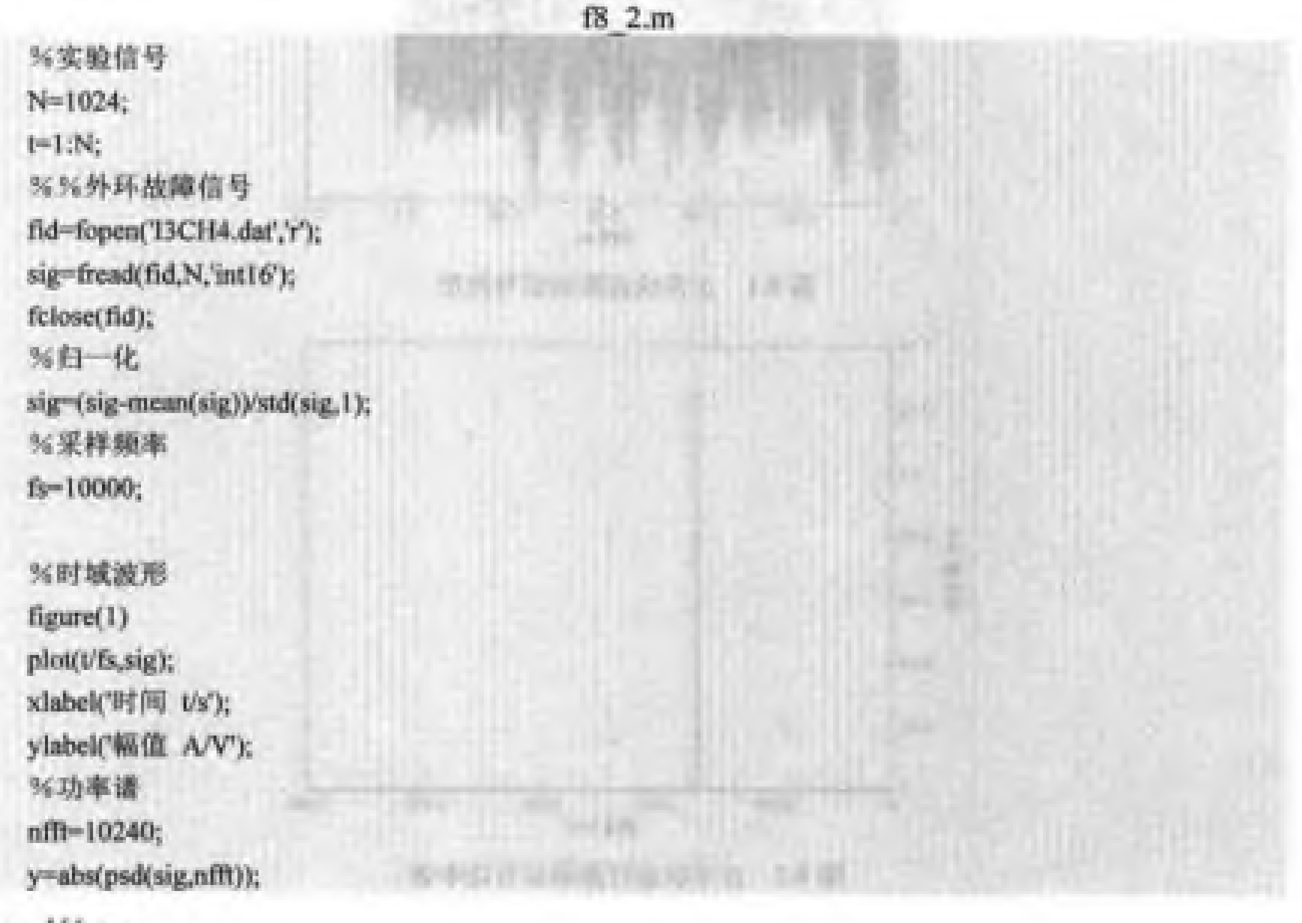


图 8.3　正常状态时振动信号 Choi-Williams 分布

8.2.2　Choi–Williams 分布用于轴承外环故障诊断

第一次的轴承故障设置为外环局部剥落。这种情况下，轴承故障特征频率为外环通过频率 f_{op} 及其倍频，理论计算得到 f_{op}=475.6Hz。

f8_2.m



```
%实验信号
N=1024;
t=1:N;
%%外环故障信号
fid=fopen('l3CH4.dat','r');
sig=fread(fid,N,'int16');
fclose(fid);
%归一化
sig=(sig-mean(sig))/std(sig,1);
%采样频率
fs=10000;

%时域波形
figure(1)
plot(t/fs,sig);
xlabel('时间 t/s');
ylabel('幅值 A/V');
%功率谱
nfft=10240;
y=abs(psd(sig,nfft));
```

```
figure(2)
plot((0:nfft/2-1)/nfft*fs,y(1:nfft/2));
xlabel('频率 f/Hz');
ylabel('功率谱 S/V');
%计算 Choi-Williams 分布
sig=hilbert(sig);
[tfr,t,f]=tfrcw(sig);
figure(3)
contour(t/fs,f(1:N)*fs,abs(tfr));
xlabel('时间 t/s');
ylabel('频率 f/Hz');

%低通滤波
wp=2*500/fs;
ws=2*600/fs;
rp=3;
rs=30;
[n wn]=buttord(wp,ws,rp,rs);
[b,a]=butter(n,wn);
sig=filter(b,a,sig);
%计算 Choi-Williams 分布
sig=hilbert(sig);
[tfr,t,f]=tfrcw(sig);
figure(4)
contour(t/fs,f(1:N)*fs,abs(tfr));
xlabel('时间 t/s');
ylabel('频率 f/Hz');
```

实验采集了轴承外环故障时的振动信号，其时域波形和功率谱分别如图8.4和图8.5所示。

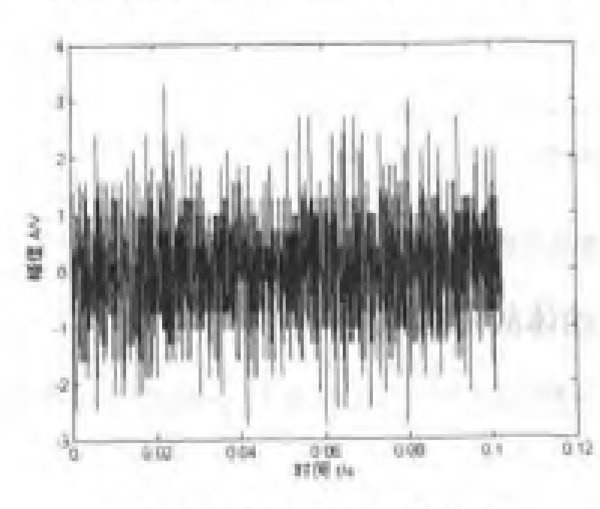




图8.4　外环故障时振动信号波形

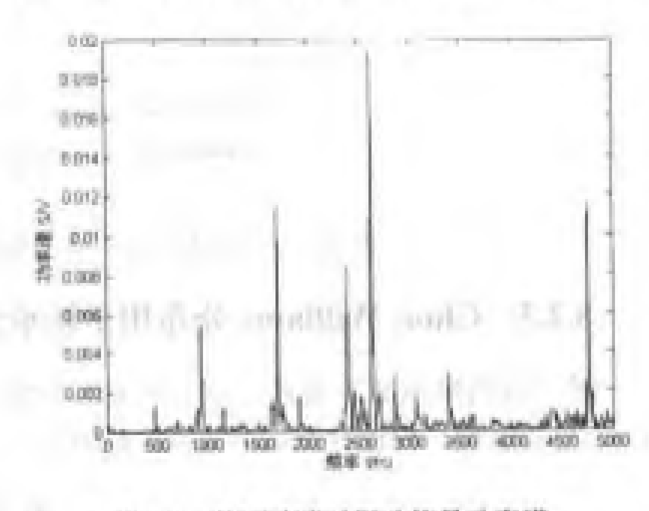


图8.5　外环故障时振动信号功率谱

由图8.5的功率谱可以看出，除了第1级齿轮的啮合频率及其谐波分量外，还存在由外

环故障导致的其他频率分量，而且外环故障特征频率也能隐约看到。计算振动信号的 Choi-Williams 分布如图 8.6 所示，其时频面与正常状态时相比也增加了很多窄带分量，但外环故障特征频率分量几乎看不到，为此对振动信号进行了低通滤波处理，滤波器截止频率为 500Hz，然后在计算其 Choi-Williams 分布，如图 8.7 所示，明显可以看出时频面存在 475Hz 的常频率分量。

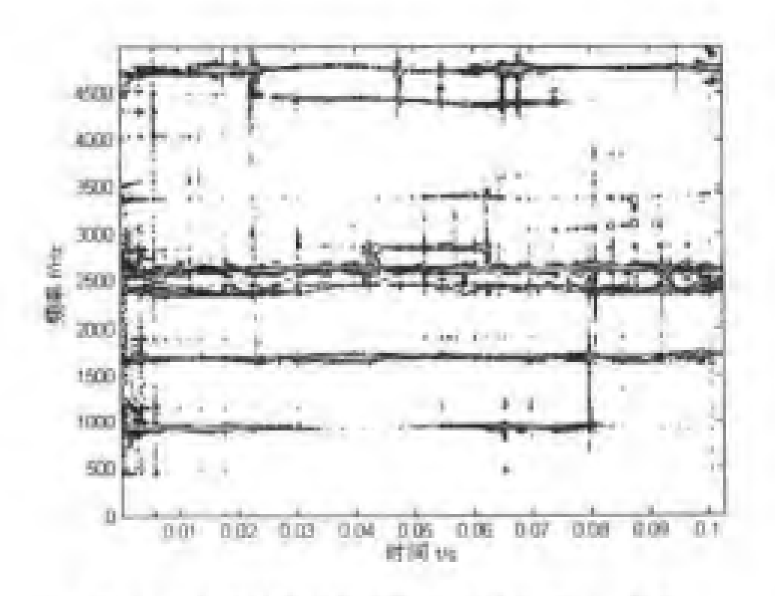


图 8.6　外环故障时振动信号 Choi-Williams 分布

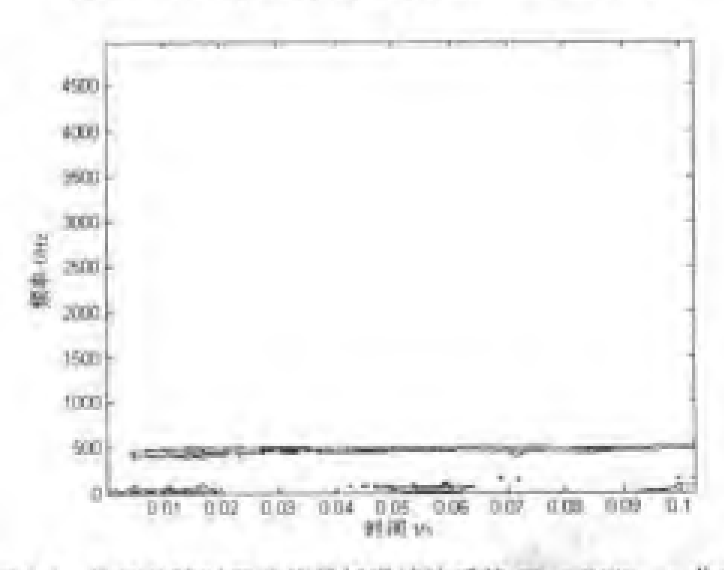


图 8.7　外环故障时振动信号低通滤波后的 Choi-Williams 分布

8.2.3　Choi–Williams 分布用于轴承滚动体故障诊断

第二次的轴承故障设置为滚动体表面点蚀。这种情况下，故障特征频率为滚动体通过频率 f_{bc} 及其倍频，理论计算得到 f_{bc}=218.98Hz。

f8_3.m

```
%实验信号
N=1024;
```

```
t=1:N;
%滚动体故障信号
fid=fopen('rCH4.dat','r');
sig=fread(fid,N,'int16');
fclose(fid);
%归一化
sig=(sig-mean(sig))/std(sig,1);
%采样频率
fs=10000;

%时域波形
figure(1)
plot(t/fs,sig);
xlabel('时间 t/s');
ylabel('幅值 A/V');
%功率谱
nfft=10240;
y=abs(psd(sig,nfft));
figure(2)
plot((0:nfft/2-1)/nfft*fs,y(1:nfft/2));
xlabel('频率 f/Hz');
ylabel('功率谱 S/V');

%计算 Choi-Williams 分布
sig=hilbert(sig);
[tfr,t,f]=tfrcw(sig);
figure(3)
contour(t/fs,f(1:N)*fs,abs(tfr));
xlabel('时间 t/s');
ylabel('频率 f/Hz');

%低通滤波
wp=2*400/fs;
ws=2*500/fs;
rp=3;
rs=30;
[n wn]=buttord(wp,ws,rp,rs);
[b,a]=butter(n,wn);
sig=filter(b,a,sig);

%计算 Choi-Williams 分布
sig=hilbert(sig);
[tfr,t,f]=tfrcw(sig);
```

```
figure(4)
contour(t/fs,f(1:N)*fs,abs(tfr));
xlabel('时间 t/s');
ylabel('频率 f/Hz');
```

实验采集了轴承滚动体故障时的振动信号，其时域波形和功率谱分别如图 8.8 和图 8.9 所示。由图 8.9 的功率谱可以看出，除了齿轮的啮合频率及其谐波分量外，还存在由滚动体故障导致的其他频率分量，而且故障特征频率基本上看不到。计算振动信号的 Choi-Williams 分布如图 8.10 所示，其时频面与正常状态时相比也增加了很多窄带分量，但外环故障特征频率分量几乎看不到，为此对振动信号进行了低通滤波处理，滤波器截止频率为 400Hz，然后在计算其 Choi-Williams 分布，如图 8.11 所示，可以发现时频面上从 0.07s 开始存在 218Hz 的频率分量。

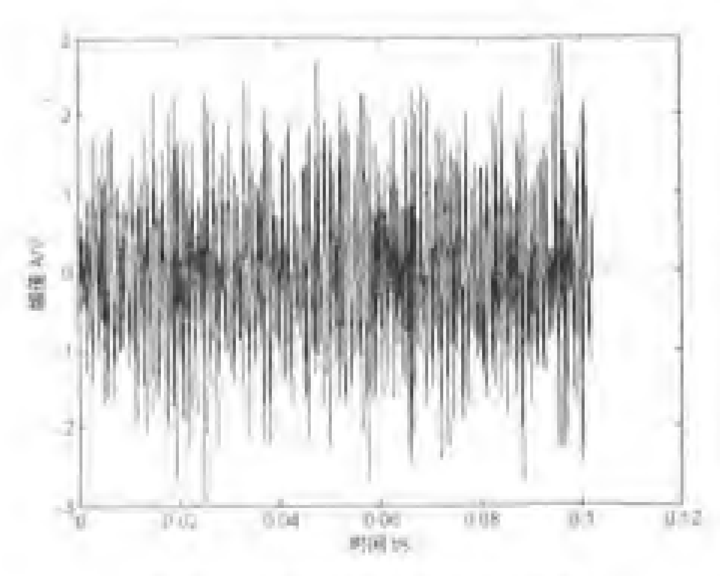


图 8.8　滚动体故障时振动信号波形

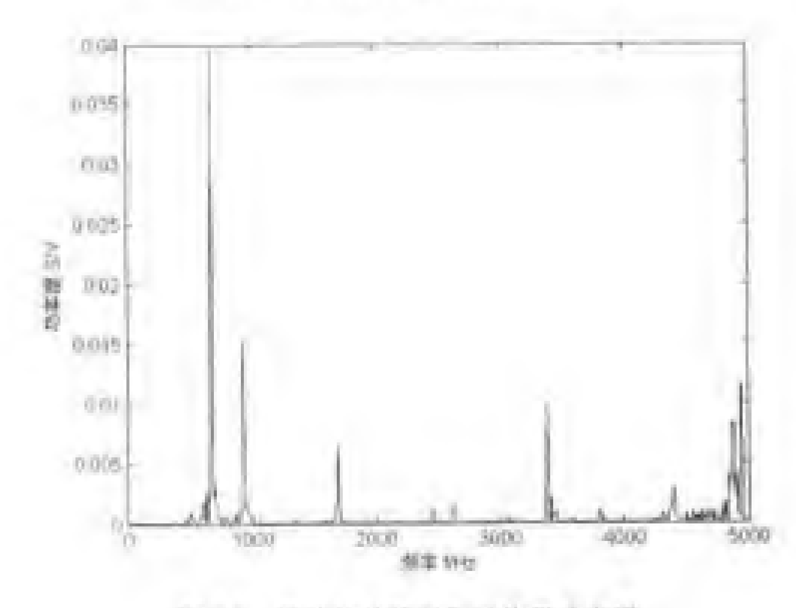


图 8.9　滚动体故障时振动信号功率谱

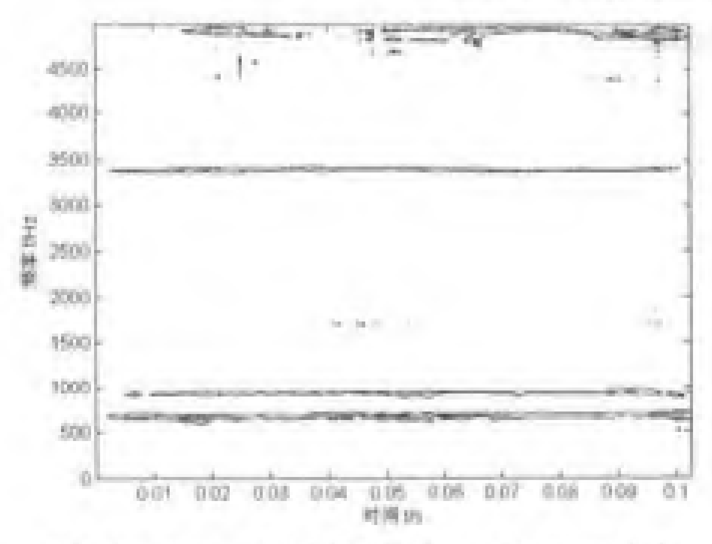


图 8.10　滚动体故障时振动信号 Choi-Williams 分布

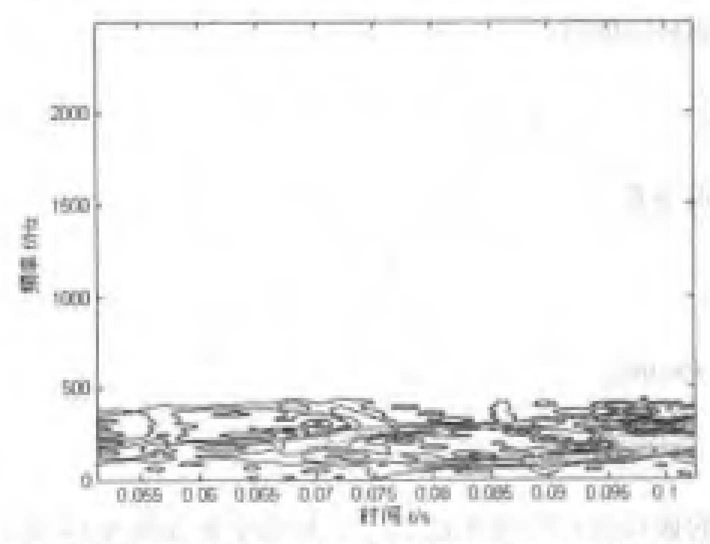


图 8.11　滚动体故障时振动信号低通滤波后 Choi-Williams 分布

8.3　时频分布在齿轮故障诊断中的应用

本节将利用伪 Wigner-Ville 分布诊断齿轮点蚀和裂纹故障。

8.3.1　正常齿轮信号的伪 Wigner–Ville 分布

f8_4.m

```
%实验信号
N=1024;
t=1:N;
fid=fopen('normal.dat','r');%正常信号
sig=fread(fid,N,'int16');
fclose(fid);
%归一化
```

```
sig=(sig-mean(sig))/std(sig,1);
%采样频率
fs=10000;

%时域波形
figure(1)
plot(t/fs,sig);
xlabel('时间 t/s');
ylabel('幅值 A/V');

%功率谱
nfft=10240;
y=abs(psd(sig,nfft));
figure(2)
plot((0:nfft/2-1)/nfft*fs,y(1:nfft/2));
xlabel('频率 f/Hz');
ylabel('功率谱 S/V');

%计算伪 Wigner-Ville 分布
sig=hilbert(sig);
[tfr,t,f]=tfrpwv(sig);
figure(3)
contour(t/fs,f(1:N)*fs,abs(tfr));
xlabel('时间 t/s');
ylabel('频率 f/Hz');
```

正常状态下采集的振动信号如图 8.12 所示，其功率谱如图 8.13 所示。由功率谱图可以看出，只存在第 1 级齿轮的啮合频率，说明正常状态下主要是齿轮的啮合信号。计算振动信号的伪 Wigner-Ville 分布如图 8.14 所示。

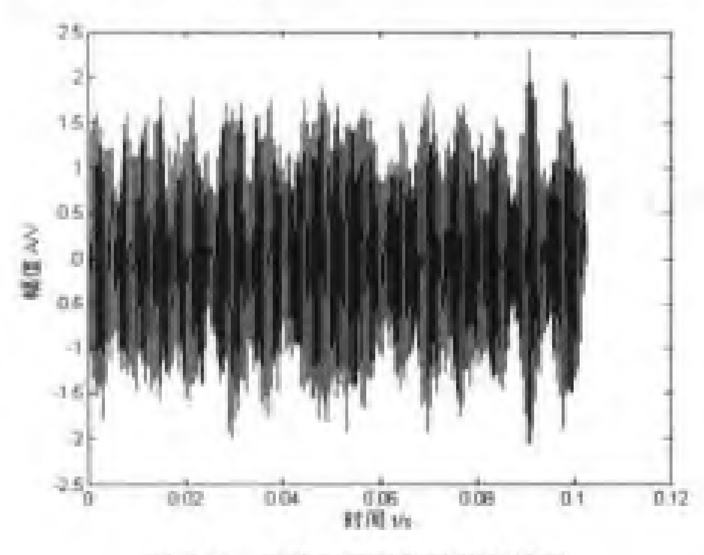


图 8.12　齿轮正常时振动信号波形

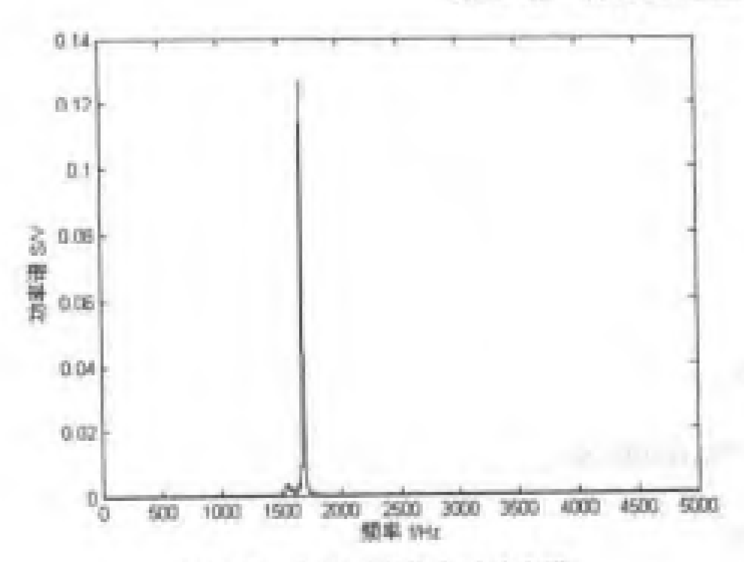


图 8.13 齿轮正常状态时功率谱

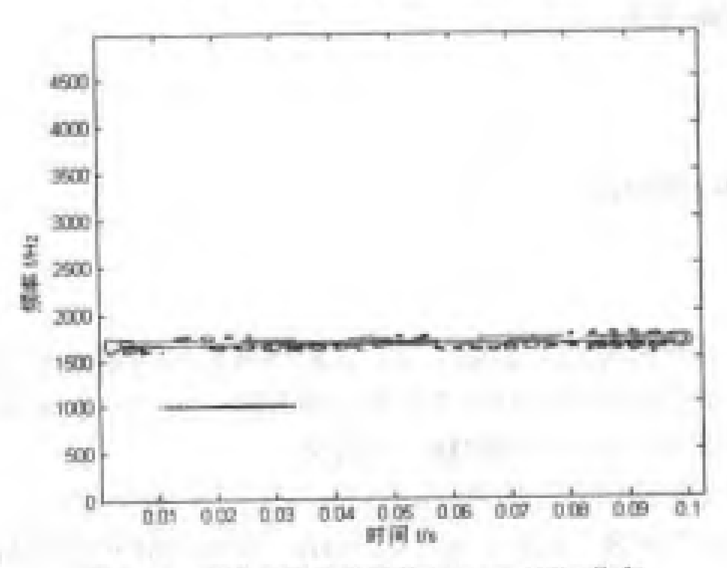


图 8.14 齿轮正常状态时伪 Wigner-Ville 分布

8.3.2 伪 Wigner-Ville 分布用于齿轮点蚀故障诊断

f8_5.m

```
%实验信号
N=1024;
t=1:N;
%齿轮点蚀故障信号
fid=fopen('gearch9.dat','r');
sig=fread(fid,N,'int16');
fclose(fid);
%归一化
sig=(sig-mean(sig))/std(sig,1);
%采样频率
fs=10000;
```

```
%时域波形
figure(1)
plot(t/fs,sig);
xlabel('时间 t/s');
ylabel('幅值 A/V');
%功率谱
nfft=10240;
y=abs(psd(sig,nfft));
figure(2)
plot((0:nfft/2-1)/nfft*fs,y(1:nfft/2));
xlabel('频率 f/Hz');
ylabel('功率谱 S/V');

%计算伪 Wigner-Ville 分布
sig=hilbert(sig);
[tfr,t,f]=tfrpwv(sig);
figure(3)
contour(t/fs,f(1:N)*fs,abs(tfr));
xlabel('时间 t/s');
ylabel('频率 f/Hz');
```

实验采集了齿轮点蚀故障时的振动信号，其时域波形和功率谱分别如图 8.15 和图 8.16 所示。从时域波形上能隐约看出信号中存在有点蚀故障引起的冲击分量，而功率谱图中频率分量主要集中在低频部分，但是不能解释点蚀故障。

为此接下来计算了振动信号的伪 Wigner-Ville 分布，如图 8.17 所示，可以看出在时频面上存在周期性的“斑点”区域，其频率约等于 58Hz，从而说明了点蚀故障在振动信号产生周期性的冲击特征。

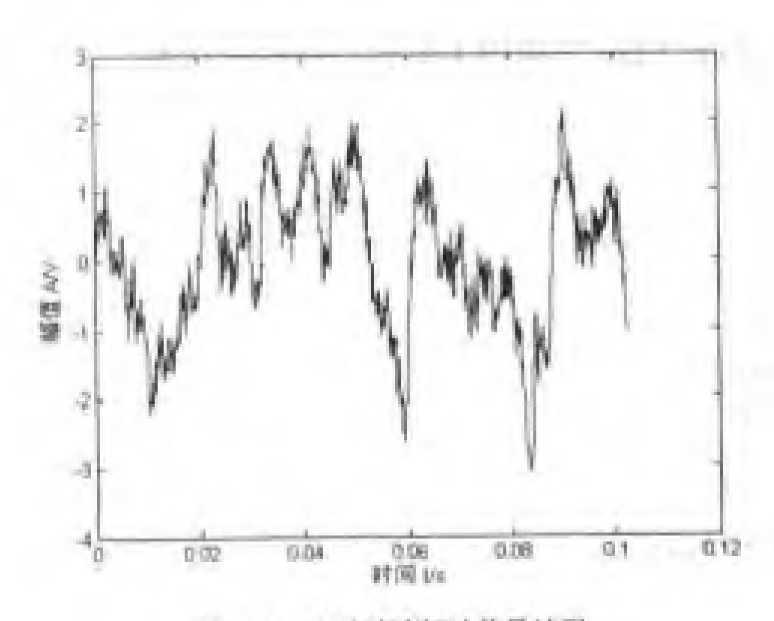


图 8.15　点蚀故障振动信号波形

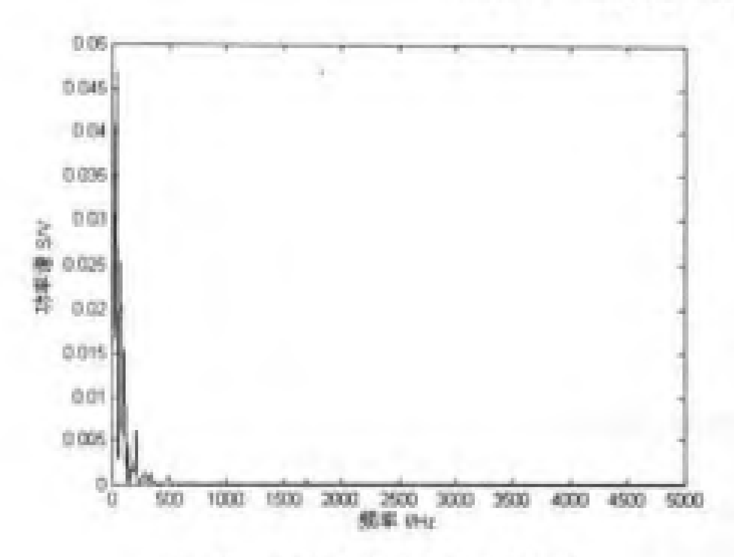


图 8.16　点蚀故障振动信号功率谱

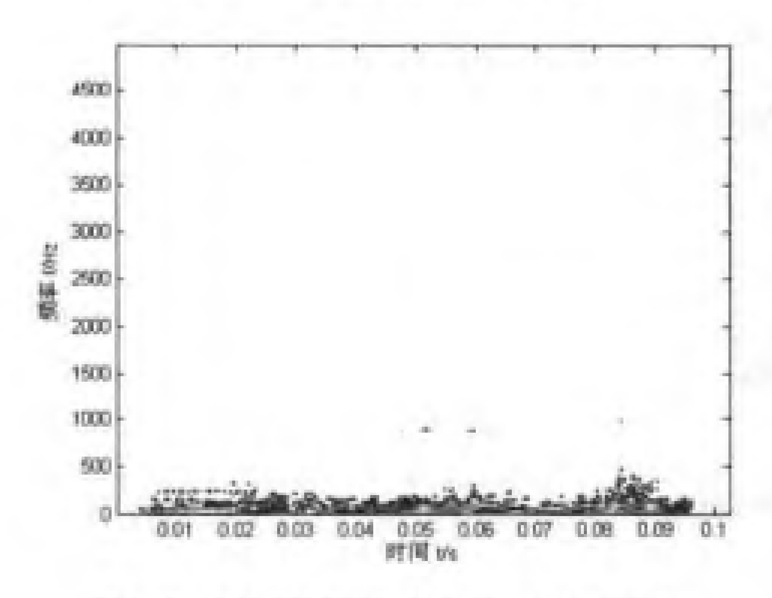


图 8.17　点蚀故障振动信号伪 Wigner-Ville 分布

8.3.3　伪 Wigner-Ville 分布用于齿轮裂纹故障诊断

f8_6.m

```
%实验信号
N=1024;
t=1:N;
%齿轮裂纹故障信号
fid=fopen('gearch1.dat','r');
sig=fread(fid,N,'int16');
fclose(fid);
%归一化
sig=(sig-mean(sig))/std(sig,1);
%采样频率
fs=10000;
```

```
%时域波形
figure(1)
plot(t/fs,sig);
xlabel('时间 t/s');
ylabel('幅值 A/V');
%功率谱
nfft=10240;
y=abs(psd(sig,nfft));
figure(2)
plot((0:nfft/2-1)/nfft*fs,y(1:nfft/2));
xlabel('频率 f/Hz');
ylabel('功率谱 S/V');
%计算伪 Wigner-Ville 分布
sig1=hilbert(sig);
[tfr,t,f]=tfrpwv(sig1);
figure(3)
contour(t/fs,f(1:N)*fs,abs(tfr));
xlabel('时间 t/s');
ylabel('频率 f/Hz');

%带通滤波
wp=2*[1500 2400]/fs;
ws=2*[1400 2500]/fs;
rp=3;
rs=30;
[n wn]=buttord(wp,ws,rp,rs);
[b,a]=butter(n,wn);
sig2=filter(b,a,sig);
%计算伪 Wigner-Ville 分布
sig=hilbert(sig2);
[tfr,t,f]=tfrpwv(sig2);
figure(4)
contour(t/fs,f(1:N)*fs,abs(tfr));
xlabel('时间 t/s');
ylabel('频率 f/Hz');
```

实验采集了齿轮裂纹故障时的振动信号，其时域波形和功率谱分别如图 8.18 和图 8.19 所示。从时域波形上不能看出齿轮裂纹故障引起的信号变化，而功率谱图中频率分量主要集中在齿轮啮合频率处，也不能解释裂纹故障。为此接下来计算了振动信号的伪 Wigner-Ville 分布，如图 8.20 所示，可以看出在啮合频率附近存在很多频率分量。

进一步对裂纹故障时的振动信号进行了带通滤波处理，然后计算其伪 Wigner-Ville 分布，如图 8.21 所示，能更清楚地看见在啮合频率 1 689Hz 和其倍频 3 400Hz 附近存在频率分量，谱线间隔约等于 4 倍转频，从而说明裂纹故障导致了调制现象的发生。

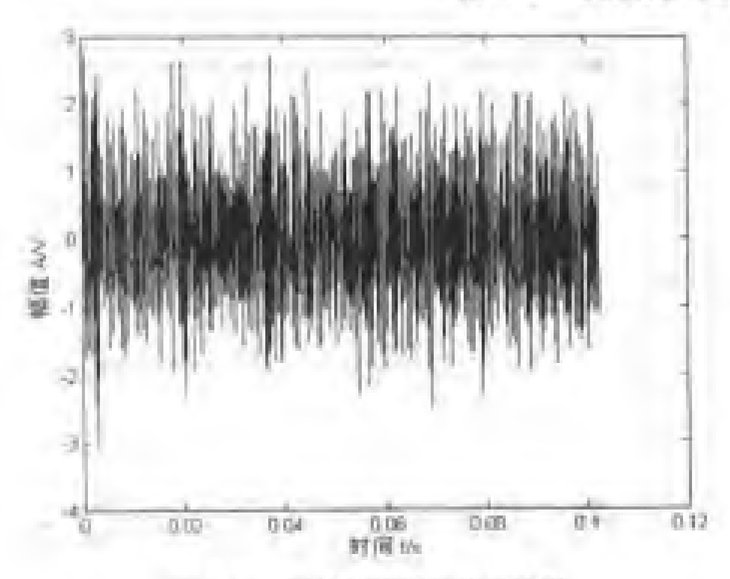


图 8.18 裂纹故障振动信号波形

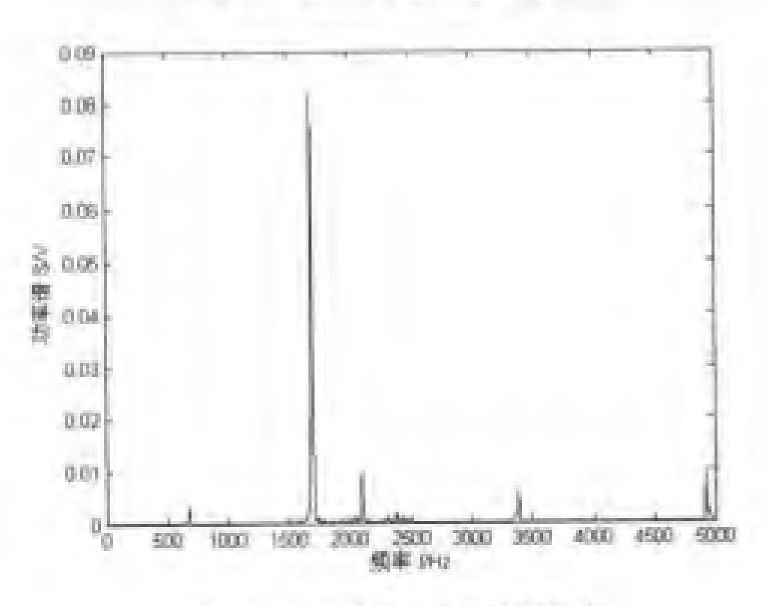


图 8.19 裂纹故障振动信号功率谱

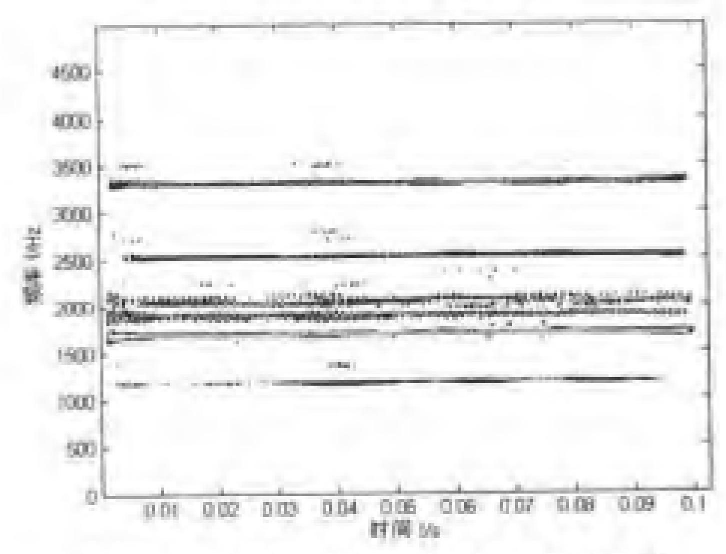


图 8.20 裂纹故障振动信号伪 Wigner-Ville 分布

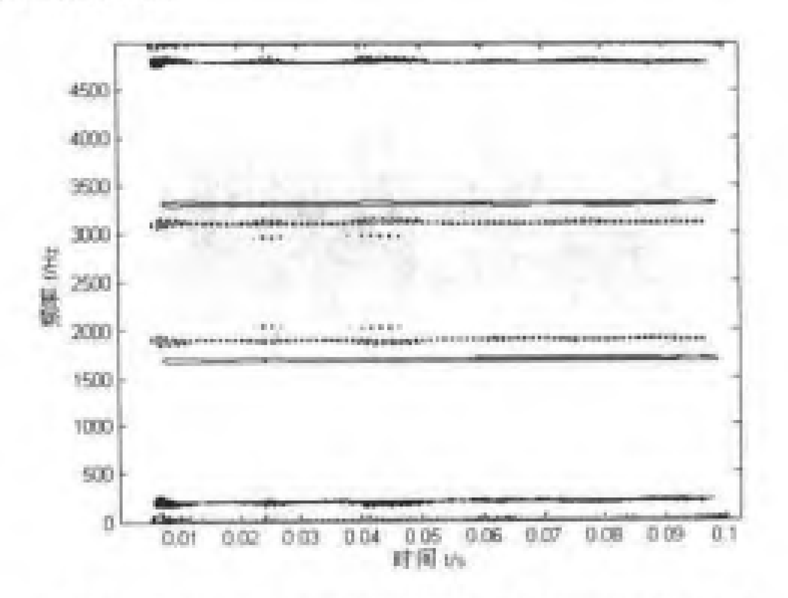


图 8.21　裂纹故障振动信号带通滤波后伪 Wigner-Ville 分布

第3部分　小波工具箱及其应用

从数学的角度来看，小波实际上是在特定空间内按照称之为小波的基函数（通常每一个基都有明确的物理意义）对数学表达式的展开和逼近。作为一种快速高效、高精度的近似方法，小波理论构成调和分析中傅立叶分析的重要发展。与傅立叶变换由三角基函数构成相比，小波基函数是一个具有快速衰减、充分光滑、能量主要集中在一个局部区域的函数经过伸缩和平移得到的函数集合。因此，从信号处理的角度来看，作为一种新的时频分析工具，小波变换克服了傅立叶变换方法表示信息时能够清晰地揭示出信号的频率特征，但不能反映时间域上的局部信息的缺陷。而局部性质的描述不论是在理论上，还是在实际应用方面都是十分重要的。利用小波进行时频分析时，由于同时具有时间和频率的局部特性以及多分辨率分析特性，故使得对非平稳信号的分析变得相对容易。

小波分析是近年来迅速发展起来的新兴学科，已经成为众多学科共同关注的热点。传统上使用傅立叶变换的地方现在都可以用小波分析代替，它较好地解决了时间和频率分辨率的矛盾。目前小波分析正速步应用于信号分析、系统控制、图像处理、量子力学、计算机识别、语音识别和合成以及故障诊断等领域。本部分的内容包括：

- Matlab 小波工具箱；
- 小波变换在信号处理中的应用；
- 小波变换在图像处理中的应用；
- 小波包在信号处理中的应用；
- 小波包在图像处理中的应用；
- 小波分析在交通领域中的应用；
- 提升小波变换的应用初步。

第 9 章　Matlab 小波工具箱

Matlab 的精华在于它提供了大量的库函数，能够直接被用户调用。对于学习小波分析来说，如果能够掌握并熟练使用小波工具箱里面的函数，实际上就掌握了该工具箱。更令人高兴的是，Matlab 还提供了小波分析的图形化方式，也就是我们常说的图形用户界面（GUI），用户不需要使用任何函数，更不需要编写任何程序，就可以形象直观地了解 Matlab 的强大小波分析功能。

本章将分类讲述这些小波库函数的基本功能和用法，同时对图形用户界面进行简单地介绍。读者在入门以后，可以结合 Matlab 软件本身提供的帮助和实例进行学习。相信通过本章的学习，读者可进一步加深对小波中各种概念的理解，并为后面学习各种小波分析函数打下较好的基础。本章主要内容：

- 小波工具箱函数简介；
- Matlab 小波分析的图形实现；
- 自定义小波基函数的添加。

9.1　小波工具箱函数简介

小波工具箱的函数按类别来分，主要有一般处理函数、小波家族重要函数、一维连续小波变换函数、一维离散小波变换函数、一维平稳小波变换函数、二维离散小波变换函数、二维平稳小波变换函数、小波降噪和压缩函数、小波包降噪和压缩函数以及树管理等其他应用函数，表 9.1～表 9.11 列出了这些函数以及各自的功能。另外，提升小波变换函数是 Matlab R14 的新功能，本书将在第 15 章集中介绍。

表 9.1　小波工具箱的一般处理函数及功能

函 数 名	功　能	函 数 名	功　能
biorfilt	双正交小波滤波器组	scal2frq	将尺度转换为频率
centfrq	计算小波中心频率	wavefun	小波和尺度函数
dyaddown	二元抽取	wavefun2	二维小波尺度函数
dyadup	二元插值	wavemngr	小波管理器
intwave	积分小波函数 ψ	wfilters	小波滤波器
orthfilt	正交小波滤波器组	wmaxlev	小波分解的最大尺度
qmf	镜像二次滤波器		

表 9.2　　小波家族重要函数及功能

函 数 名	功 能	函 数 名	功 能
biorwavf	双正交样条小波滤波器	mexihat	Mexican hat 小波
cgauwavf	复高斯小波	meyer	Meyer 小波
cmorwavf	复 Morlet 小波	meyeraux	Meyer 小波辅助函数
coifwavf	Coiflet 小波滤波器	morlet	Morlet 小波
dbaux	计算 Daubechies 小波滤波器	rbiowavf	反双正交样条小波滤波器
dbwavf	Daubechies 小波滤波器	shanwavf	复 Shannon 小波
fbspwavf	复频率样条小波	symaux	计算 Symlet 小波
gauswavf	Gaussian 小波	symwavf	Symlets 小波滤波器

表 9.3　　一维连续小波变换函数

函 数 名	功 能	函 数 名	功 能
pat2cwav	由模式构造小波	cwt	一维连续小波变换函数

表 9.4　　一维离散小波变换函数

	函 数 名	说 明
分解函数	dwt	一维单尺度离散小波变换
	wavedec	一维多尺度小波分解
	wmaxlev	小波分解的最大尺度
合成重构函数	idwt	一维单尺度小波逆变换
	waverec	一维多尺度小波重构
	wrcoef	对一维小波系数进行单支重构
	upcoef	一维小波系数的直接重构
分解结构工具	detcoef	提取一维小波细节系数
	appcoef	提取一维小波近似系数
	upwlev	一维单尺度小波分解
	dwtmode	离散小波变换扩展模式

表 9.5　　一维平稳小波变换函数

函 数 名	功 能	函 数 名	功 能
swt	一维离散平稳小波变换	iswt	一维离散平稳小波逆变换

表 9.6　　二维离散小波变换函数

	函 数 名	说　明
分解函数	dwt2	二维单尺度小波变换
	wavedec2	多尺度二维小波分解（二维多分辨率分析函数）
	wmaxlev	允许的最大尺度分解
合成重构工具	idwt2	二维单尺度小波逆变换
	waverec2	二维多尺度小波重构
	wrcoef2	对二维小波系数进行单支重构
	upcoef2	二维小波系数的直接重构
分解结构工具	detcoef2	提取二维小波分解高频系数
	appcoef2	提取二维小波分解低频系数
	upwlev2	二维小波分解的单尺度重构

表 9.7　　二维平稳小波变换函数

函 数 名	功　能	函 数 名	功　能
swt2	二维离散平稳小波变换	iswt2	二维离散平稳小波反变换

表 9.8　　小波包分析函数

	函 数 名	说　明
分解函数	wpcoef	小波包系数
	wpdec	一维小波包分解
	wpdec2	二维小波包分解
	wpsplt	分割（分解）小波包
合成重构函数	wprcoef	重构小波包系数
	wprec	一维小波包全重构
	wprec2	二维小波包分解
	wpjoin	重新组合小波包
分解结构工具	besttree	最优小波包树
	bestlevt	最优完整小波包树
	entrupd	更新小波包的熵值
	get	得到 WPTREE 对象的内容
	read	读取 WPTREE 对象的值
	wentropy	计算小波包的熵值
	wp2wtree	从小波包树中提取小波树
	wpcutree	截除小波包分解树
	wenergy2	计算二维小波分解能量

表 9.9 小波降噪和压缩函数

函 数 名	说 明	函 数 名	说 明
wnoise	产生小波的噪声测试数据	wdcbm2	使用 Birge-Massart 策略的二维小波阈值
wnoisest	估计一维小波系数的标准差	wthcoef	一维小波系数阈值处理
ddencmp	获取降噪或压缩的默认值	wthcoef2	二维小波系数阈值处理
wthresh	进行软阈值或硬阈值处理	wden	使用小波进行一维自动降噪
thselect	选择降噪时的阈值	wdencmp	使用小波进行降噪或压缩
wbmpen	设置一维或二维小波降噪的阈值	wthrmngr	阈值设置管理
wdcbm	使用 Birge-Massart 策略的一维小波阈值		

表 9.10 小波包降噪和压缩函数

函 数 名	说 明	函 数 名	说 明
ddencmp	获取降噪或压缩的默认值	wpdencmp	使用小波包进行降噪和压缩
wpbmpen	小波包降噪的 Penalized 阈值	wthrmngr	阈值设置管理
wpthcoef	小波包系数阈值处理		

表 9.11 小波工具箱其他应用函数

函 数 名	说 明	函 数 名	说 明
wfbm	分数布朗运动合成	ntree	构造类 NTREE（N-ary Tree）
wfbmesti	分数布朗运动参数估计	get	获取树对象域内容
wfusmat	矩阵或数组融合	read	读取树对象域值
wfusimg	融合两幅图像	set	设置树对象域内容
wptree	构造小波包树（WPTREE）类	write	对树对象域赋值
dtree	构造类 DTREE（Data Tree）	plot	树管理图形工具

9.1.1 小波通用函数

在接触小波各种分析函数之前，有必要了解一下各种通用函数，小波工具箱 3.0 版本中共提供了 13 个小波通用函数，以下做简单介绍。

1. biorfilt 函数

【函数功能】双正交小波滤波器组。

【语法格式】

① [Lo_D,Hi_D,Lo_R,Hi_R] = biorfilt(DF,RF)

② [Lo_D1,Hi_D1,Lo_R1,Hi_R1,Lo_D2,Hi_D2,Lo_R2,Hi_R2] = biorfilt(DF,RF,'8')

【使用说明】通过调用该函数可以产生与双正交小波相关联的 4 个或 8 个滤波器组。

- DF——分解滤波器。
- RF——重构滤波器。

格式①用来计算该小波的 4 个滤波器。DF 和 RF 将确定了双正交小波。

- Lo_D——分解低通滤波器。
- Hi_D——分解高通滤波器。
- Lo_R——重构低通滤波器。
- Hi_R——重构高通滤波器。

格式②返回 8 个滤波器，包括 4 个分解小波相关的滤波器和 4 个重构小波相关的滤波器。

2. centfrq 函数

【函数功能】计算小波中心频率。

【语法格式】

① FREQ = centfrq('wname')

② FREQ = centfrq('wname',ITER)

③ [FREQ,XVAL,RECFREQ] = centfrq('wname',ITER,'plot')

调用①函数得到'wname'小波的中心频率。

格式②中，ITER 是被 wavefun 函数重复调用的次数，wavefun 用来计算小波。

格式③返回的是基于 2^{ITER} 点网格 XVAL 上的近似 RECFREQ，计算相关的中心频率，并且画出小波函数和 RECFREQ。

3. dyaddown 函数

【函数功能】二元抽取。

【语法格式】

① Y = dyaddown(X,EVENODD)

② Y = dyaddown(X)

③ Y = dyaddown(X,EVENODD,'type')

④ Y = dyaddown(X,'type',EVENOD)

【使用说明】格式①用于从向量 X 中每隔一个元素抽取一元素组成向量 Y。至于 Y 是包含 X 的奇数元素还是偶数元素，取决于 EVENODD 的值：

- 如果 EVENODD 为偶数，则进行偶抽取，Y(k)=X(2k)。
- 如果 EVENODD 为奇数，则进行奇抽取，Y(k)=X(2k+1)。

格式②省略了 EVENODD 值，相当于格式①中的 EVENODD＝0，即等同于 Y=dyaddown (X,0)。

格式③、④中，X 是一个矩阵：

- 如果'type' = 'c'，则抽取矩阵 X 的列。
- 如果'type' = 'r'，则抽取矩阵 X 的行。
- 如果'type' = 'm'，则抽取矩阵 X 的行和列。

EVENOD 的含义上面的解释一样有效。

如果忽略了EVENODD或者'type'的定义，dyaddown函数将默认EVENODD=0（奇采样）和'type' = 'c'（列采样），即Y=dyaddown(X)等同于Y=dyaddown(X,0,'c')；Y=dyaddown(X,'type')等同于Y=dyaddown(X,0,'type')；Y=dyaddown(X,EVENODD)等同于Y=dyaddown(X,EVENODD,'c')。

4. dyadup 函数

【函数功能】二元插值。

【语法格式】

① Y = dyadup(X,EVENODD)

② Y = dyadup(X)

③ Y = dyadup(X,EVENODD,'type')

④ Y = dyadup(X,'type',EVENODD)

【使用说明】该函数用来从向量X中每隔一个元素插入一个0元素，组成向量Y，这在小波重构算法中是非常必要的。

格式①中:

- 如果EVENODD为偶数，则进行偶插值，即Y(2k−1) = X(k), Y(2k) = 0。
- 如果EVENODD为奇数，则进行奇插值，即Y(2k−1) = 0, Y(2k) = X(k)。

Y = dyadup(X)等同于Y = dyadup(X,1)（偶插值）。

对于格式③和④，X是一个矩阵：

- 如果'type' = 'c'，则插入列。
- 如果'type' = 'r'，则插入行。
- 如果'type' = 'm'，则插入行和列。

③和④中参数EVENODD的意义如同上面的介绍。

5. intwave 函数

【函数功能】积分小波函数ψ。

【语法格式】

① [INTEG,XVAL] = intwave('wname',PREC)

② [INTEG,XVAL] = intwave('wname',PREC,PFLAG)

③ [INTEG,XVAL] = intwave('wname')

【使用说明】计算小波函数ψ在区间（$-\infty$, XVAL）上的积分INTEG，即$\int_{-\infty}^{x}\psi(y)\mathrm{d}y$。其中$x$位于XVAL。

函数ψ是在2^{PREC}点网格XVAL上的近似，PREC是正整数。'wname'是包含小波函数ψ名称的字符串。输出变量INTEG是实数还是复数，主要取决于小波类型。对于双正交小波，[INTDEC,XVAL,INTREC] = intwave('wname',PREC)计算小波分解函数积分ψ_{dec}和小波重构函数ψ_{dec}的积分，结果分别为INTDEC和INTREC。

[INTEG,XVAL] =intwave('wname',PREC)等同于[INTEG,XVAL]= intwave('wname',PREC,0)。

[INTEG,XVAL] = intwave('wname')等同于[INTEG,XVAL] = intwave('wname',8)。

当使用三变量函数intwave('wname',IN2,IN3)时, PREC = max(IN2,IN3) 并画图。

当 IN2＝0 时，intwave('wname',0) 等同于 intwave('wname',8,IN3)。

intwave('wname') 等同于 intwave('wname',8)。

intwave 只能用于连续小波变换。

6. orthfilt 函数

【函数功能】正变小波滤波器组。

【语法格式】[Lo_D,Hi_D,Lo_R,Hi_R] = orthfilt(W)

【使用说明】该函数用来计算与小波函数对应的尺度滤波器相关的 4 个滤波器。其中：

- Lo_D——分解低通滤波器。
- Hi_D——分解高通滤波器。
- Lo_R——重构低通滤波器。
- Hi_R——重构高通滤波器。

7. qmf 函数

【函数功能】镜像二次滤波器。

【语法格式】

① Y = qmf(X,P)

② Y = qmf(X)

【使用说明】对于格式①，当 P 为偶数时，函数改变向量 X 中偶数位置的元素符号；当 P 为奇数时，同样改变奇数位置的元素符号。Y = qmf (X)等同于 Y = qmf (X,0)。

8. scal2frq 函数

【函数功能】将尺度转换为频率。

【语法格式】F = scal2frq(A,'wname',DELTA)

【使用说明】该函数返回由尺度 A，小波函数'wname'和采样周期 DELTA 决定的准频率。

格式 scal2frq(A,'wname')等同于 scal2frq(A,'wname',1)。

9. wavefun 函数

【函数功能】小波和尺度函数。

【语法格式】

① [PHI,PSI,XVAL] = wavefun('wname',ITER)

② [PHI1,PSI1,PHI2,PSI2,XVAL] = wavefun('wname',ITER)

③ [PSI,XVAL] = wavefun('wname',ITER)

④ [...] = wavefun('wname',A,B)

【使用说明】该函数返回小波函数 wname 和对应尺度函数的近似值（如果存在）。正整数 ITER 决定了重复计算的次数，从而决定了近似值的精确程度。

对于正变小波，格式①返回尺度函数和小波函数。

对于双正变小波，格式②返回分别用于分解和重构的尺度函数和小波函数，分解的为 (PHI1,PSI1)，重构的为 (PHI2,PSI2)。

对于 Meyer 小波，有[PHI,PSI,XVAL] = wavefun('wname',ITER)。

对于没有尺度函数的小波，如 Morlet 小波、Mexican Hat 小波、Gaussian derivatives 小波和复小波，使用格式③。

格式④中，A 和 B 都是正整数，格式④等价于[...] = wavefun('wname',max（A,B）)，并且画图。

当 A=0 时，[...]=wavefun('wname',0)等价于[...]=wavefun('wname',8,0)，[...]=wavefun('wname')等价于[...] = wavefun('wname',8)。

10. wavefun2 函数

【函数功能】二维小波尺度函数。

【语法格式】

① [S,W1,W2,W3,XYVAL] = wavefun2('wname',ITER)

② [S,W1,W2,W3,XYVAL] = wavefun2('wname',ITER,'plot')

③ [S,W1,W2,W3,XYVAL] = wavefun2('wname',A,B)

【使用说明】对于正交小波，wavefun2 返回尺度函数和 3 个小波函数，它们是一维小波函数和尺度函数的矢量积。

对于[PHI,PSI,XVAL] = wavefun('wname',ITER)，尺度函数 S 是 PHI 和 PSI 的矢量积。小波函数 W1、W2 和 W3 分别是（PHI,PSI)、（PHI,PSI）以及（PSI,PSI）对应的矢量积。

二维变量 XTVAL 是（XVAL,XVAL）矢量积得到的 $2^{ITER}\times2^{ITER}$ 点网格。正整数 ITER 决定了重复计算的次数，因此决定近似精度。

[S,W1,W2,W3,XYVAL] = wavefun2('wname',ITER,'plot')计算并画出这些函数。

对于格式②，A 和 B 都是正整数，等同于[S,W1,W2,W3,XYVAL] = wavefun2('wname',max(A,B))。它计算小波函数和尺度函数的近似值并画图。

当 A 被设定为 0 时，[S,W1,W2,W3,XYVAL]=wavefun2('wname',0)等同于 [S,W1,W2,W3,XYVAL]=wavefun2('wname',4,0)。

[S,W1,W2,W3,XYVAL] = wavefun2('wname')等同于[S,W1,W2,W3,XYVAL]=wavefun2('wname',4)。

11. wavemngr 函数

【函数功能】小波管理器。

【语法格式】

① wavemngr('add',FN,FSN,WT,NUMS,FILE)

② wavemngr('add',FN,FSN,WT,NUMS,FILE,B)

③ wavemngr('del',N)

④ wavemngr('restore')

⑤ wavemngr('restore',IN2)

⑥ OUT1 = wavemngr('read')

⑦ OUT1 = wavemngr('read',IN2)

⑧ OUT1 = wavemngr('read_asc')

【使用说明】该函数主要用于小波管理。它允许添加、删除、存储和读取小波。

格式①或②或者 wavemngr('add',FN,FSN,WT,{NUMS,TYPNUMS},FILE) 或 wavemngr('add',FN,FSN,WT,{NUMS,TYPNUMS},FILE,B)，都可以增加小波函数到工具箱中。

这里 FN 是 Family Name 的缩写；FSN 是 Family Short Name 的缩写。

WT 定义了小波类型：WT = 1 表示正交小波，WT = 2 表示正交小波，WT = 3 表示带尺度函数的小波，WT = 4 表示无尺度函数的小波，WT = 5 表示无尺度函数的复小波。

如果小波族中只有一个小波，则 NUMS = ''。例如 mexh 和 morl。

如果小波族包含多个小波，NUMS 表示小波参数的字符串，它列出了各个元素，中间用空格隔开。例如 bior: NUMS = '1.1 1.3 ... 4.4 5.5 6.8'。

如果小波是无限小波族中的一部分，那么 NUMS 表示小波参数的字符串，它列出了各个元素，中间用空格隔开，并以**结束。例如：

db: NUMS = '1 2 3 4 5 6 7 8 9 10 **'

shan: NUMS = '1-1.5 1-1 1-0.5 1-0.1 2-3 **'

对于后两种情况，TYPNUMS 小波参数的输入格式为：'integer'或者'real'或者'string'，默认值是'integer'。例如：db: TYPNUMS = 'integer'，bior: TYPNUMS = 'real'，shan: TYPNUMS = 'string'。

FILE 为 MAT 文件或者 M 文件的名称。

对于 WT 为 3、4 或 5 的情形，B=[lb ub]，它指定小波有效支撑的上、下边界。

格式③从小波族中删除一个小波。N 是字符串，表示 FSN 或者小波名称。

格式④、⑤采用 ASCII 文件来保存原始的小波，这里 IN2 是虚拟变量。

格式⑥返回小步族的名称。

格式⑦返回所有的小波名称，IN2 是虚拟变量。

格式⑧读取 wavelets.asc 文件并且返回小波信息。

12. wfilters 函数

【函数功能】小波滤波器。

【语法格式】

① [Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('wname')

② [F1,F2] = wfilters('wname','type')

格式①用来计算正交小波或双正交小波 wname 相关的 4 个滤波器。格式②返回以下滤波器：

- 如果'type' = 'd'，则返回分解滤波器 Lo_D 和 Hi_D。
- 如果'type' = 'r'，则返回重构滤波器 Lo_R 和 Hi_R。
- 如果'type' = 'l'，则返回低通滤波器 Lo_D 和 Lo_R。
- 如果'type' = 'h'，则返回高通滤波器 Hi_D 和 Hi_R。

13. wmaxlev 函数

【函数功能】小波分解的最大尺度。

【语法格式】L = wmaxlev (S,'wname')

【使用说明】该函数是一维或者二维小波函数或小波包的导向函数，它返回信号或者图像的最大分解尺度，它可以帮助避免分解时超过这个值。

一般情况下，小波分解尺度都小于理论上的最大值。通常一维分解时分解尺度为 5，而二维分解尺度为 3。

9.1.2 小波函数

Matlab 7.0 的小波工具箱 3.0 系统提供了多个小波函数，用户还可以通过命令 waveinfo 获取关于这些已有小波函数的特性介绍。

1. biorwavf 函数

【函数功能】双正交样条小波滤波器。

【语法格式】[RF,DF] = biorwavf (W)

【使用说明】该函数返回与指定双正交小波 W 相关联的两个尺度滤波器。W = 'biorNr.Nd'，其中 Nr 和 Nd 的取值如表 9.12 所示。

表 9.12 Nr 和 Nd 取值表

Nr	Nd	Nr	Nd
1	1，3，5	4	4
2	2，4，6，8	5	5
3	1，3，5，7，9	6	8

输出变量：RF 是重构滤波器，DF 是分解滤波器。

2. cgauwavf 函数

【函数功能】复高斯小波。

【语法格式】[PSI,X] = cgauwavf(LB,UB,N,P)

【使用说明】此函数定义在区间[LB, UB]上的 N 点均匀分布的网络上，返回 p 次复高斯函数 $F = C_p e^{-ix} e^{-x^2}$ 。

如果 $p>8$，需要扩展符号工具箱。

输出变量是在网格 X 上计算的小波函数 PSI。[PSI,X]= cgauwavf(LB,UB,N)等同于[PSI,X]= cgauwavf(LB,UB,N,1)。

该函数的有效支撑为[–5,5]。

3. cmorwavf 函数

【函数功能】复 Morlet 小波。

【语法格式】[PSI,X] = cmorwavf (LB,UB,N,FB,FC)

【使用说明】该函数返回复 Morlet 小波，它是在 N 点均匀分布的区间[LB,UB]上由正带宽参数 FB、小波中心频率 FC 及表达式 $PSI(X) = ((\pi * FB)^{-0.5} * e^{2*i*\pi*FC*x} * e^{-x^2/FB}$ 定义。

输出变量是网格 X 上计算得到的小波函数 PSI。

4. coifwavf 函数

【函数功能】Coiflet 小波滤波器。

【语法格式】F = coifwavf (W)

【使用说明】该函数返回由 W 指定的 Coiflet 小波的尺度滤波器。其中 W = 'coifN'，N 的取值可以是 1，2，3，4，5。

5. dbaux 函数

【函数功能】计算 Daubechies 小波滤波器。

【语法格式】

① W = dbaux (N,SUMW)

② W = dbaux (N)

【使用说明】格式①返回 N 阶 Daubechies 尺度滤波器，sum(W) = SUMW。N 的可能取值为 1，2，3，……。

注意：N 太大时可能造成不稳定。

W = dbaux (N)等同于 W = dbaux (N,1)，W = dbaux(N,0)等同于 W = dbaux (N,1)。

6. dbwavf 函数

【函数功能】Daubechies 小波滤波器。

【语法格式】F = dbwavf(W)

【使用说明】该函数返回与所指定的 Daubechies 小波相关联的尺度滤波器。W = 'dbN'，N 的取值可以是：1，2，3，……，45。

7. fbspwavf 函数

【函数功能】复频率 B 样条小波。

【语法格式】[PSI,X] = fbspwavf(LB,UB,N,M,FB,FC)

【使用说明】该函数返回 M 阶复频率 B 样条小波（M≥1），带宽为 FB，小波中心频率为 FC。

函数 PSI 定义在 N 点均匀分布的区间[LB,UB]上，使用以下精确表达式计算：

PSI(X) = (FB^0.5)*((sinc(FB*X/M).^M).*exp(2*i*pi*FC*X))，其中 FB>0，FC>0。

输出变量是在网格 X 上计算得到的小波函数 PSI。

8. gauswavf 函数

【函数功能】Gaussian 小波。

【语法格式】[PSI,X] = gauswavf (LB,UB,N,P)

【函数说明】此函数定义在区间[LB, UB]上的 N 点均匀分布网络上，返回 p 次复高斯函数 $F = C_p e^{-x^2}$ 。

对于 p >8，需要扩展符号工具箱。

输出变量为在 X 上的小波函数 PSI。

[PSI,X] = gauswavf (LB,UB,N)等价于[PSI,X] = gauswavf (LB,UB,N,1)。

这些小波的有效支撑为[−5,5]。

9. mexihat 函数

【函数功能】Mexican hat 小波。

【语法格式】[PSI,X] = mexihat(LB,UB,N)

【使用说明】该函数返回有效支撑为[LB,UB]，N 点均匀分布网格上的 Mexican hat 小波。

输出变量是网格 X 上计算得到的小波函数 PSI。

该小波的有效支撑为[−5,5]。

$$\psi(x) = (\frac{2}{\sqrt{3}}\pi^{-1/4})(1-x^2)e^{-x^2/2}$$

这个函数成比例于高斯概率密度函数的二次派生函数。

10. meyer 函数

【函数功能】Meyer 小波。

【语法格式】

① [PHI,PSI,T] = meyer (LB,UB,N)

② [PHI,T] = meyer (LB,UB,N,'phi')

③ [PSI,T] = meyer (LB,UB,N,'psi')

【函数说明】

格式①返回有效支撑为[LB,UB]，N 个均匀分布的网格上估计得到的 Meyer 尺度和小波函数。N 必须为 2 的整数次幂。

输出变量是网格 T 上计算得到的尺度函数 PHI 和小波函数 PSI。这些函数的有效支撑为[−8,8]。

如果只需要 1 个函数，则允许有第 4 个变量：

[PHI,T] = meyer (LB,UB,N,'phi')

[PSI,T] = meyer (LB,UB,N,'psi')

如果使用了第 4 个变量，但不等于'phi'或者'psi'，则输出和前面的一样。

小波函数：

$\frac{2\pi}{3} \leqslant |\omega| \leqslant \frac{4\pi}{3}$ 时，$\hat{\psi}(\omega) = (2\pi)^{-1/2} e^{i\omega/2} \sin(\frac{\pi}{2} v(\frac{3}{2\pi}|\omega|-1))$

$\frac{4\pi}{3} \leqslant |\omega| \leqslant \frac{8\pi}{3}$ 时，$\hat{\psi}(\omega) = (2\pi)^{-1/2} e^{i\omega/2} \cos(\frac{\pi}{2} v(\frac{3}{4\pi}|\omega|-1))$

$|\omega| \notin [\frac{2\pi}{3}, \frac{8\pi}{3}]$ 时，$\hat{\psi}(\omega) = 0$

这里 $v(a) = a^4(35-84a+70a^2-20a^3)$，$a \in [0,1]$

尺度函数：

$|\omega| \leqslant \frac{2\pi}{3}$ 时，$\hat{\varphi}(\omega) = (2\pi)^{-1/2}$

$\frac{2\pi}{3}\leqslant|\omega|\leqslant\frac{4\pi}{3}$时，$\hat{\psi}(\omega)=(2\pi)^{-1/2}\cos(\frac{\pi}{2}\nu(\frac{3}{2\pi}|\omega|-1))$

$|\omega|>\frac{4\pi}{3}$时，$\hat{\psi}(\omega)=0$

但是如果改变辅助函数（参见 meyeraux 函数可以获得更多的信息），就可以得到不同的小波函数族。

11. meyeraux 函数

【函数功能】Meyer 小波辅助函数。

【语法格式】Y = meyeraux(X)

【使用说明】该函数返回 Meyer 小波使用的辅助函数在向量或矩阵 X 上各点的值。

该函数是：$35x^4-84x^5+70x^6-20x^7$

12. morlet 函数

【函数功能】Morlet 小波。

【语法格式】[PSI,X] = (LB,UB,N)

【使用说明】该函数返回 Morlet 小波在 N 点均匀分布的支撑[LB,UB]上的值。输出变量是在网格 X 上计算得到的小波函数 PSI。

该小波有效支撑为[−4,4]。

$$\psi(x)=\mathrm{e}^{-x^2/2}\cos(5x)$$

13. rbiowavf 函数

【函数功能】反双正交样条滤波器。

【语法格式】[RF, DF]= rbiowavf (W)

【使用说明】该函数返回由字符串 W 定义的和双正交小波相关的两个尺度滤波器。W = 'rbioNr.Nd'，这里 Nr 和 Nd 的小波函数可能取值。

输出变量：RF 是重构滤波器，DF 是分解滤波器。

14. shanwavf 函数

【函数功能】复 Shannon 小波。

【语法格式】[PSI,X] = shanwavf(LB,UB,N,FB,FC)

【使用说明】该函数返回复 Shannon 小波的值，该小波是在区间[LB,UB]上的 N 点均匀分布的网格上，由带宽 FB，小波中心频率 FC 和表达式 PSI(X) = (FB^0.5)*(sinc(FB*X).*exp(2*i*pi*FC*X))所定义。FB>0，FC>0。

输出变量是网格 X 上的小波函数 PSI。

15. symaux 函数

【函数功能】计算 Symlet 小波。

【语法格式】

① W = SYMAUX(N,SUMW)

② W = SYMAUX(N)

【使用说明】

Symlets 是最小不对称（"least asymmetric"）Daubechies 小波。

格式①是 N 阶 Symlet 尺度滤波器，并且 SUM(W) = SUMW。N 可能的取值为 1，2，3，……。

W = SYMAUX(N)等同于 W = SYMAUX(N,1)。

W = SYMAUX(N,0)等同于 W = SYMAUX(N,1)。

16. symwavf 函数

【函数功能】Symlet 小波滤波器。

【语法格式】F = symwavf(W)

【使用说明】该函数返回由字符串 W 定义的 symlet 小波相关的尺度滤波器，W= 'symN'，N 可能的取值为 2，3，……，45。

9.1.3 一维连续小波变换的函数简介

一维连续小波变换共有 2 个函数：pat2cwav 和 cwt。

1. pat2cwav 函数

【函数功能】由模式构造小波。

【语法格式】

[PSI,XVAL,NC] = pat2cwav(YPAT,METHOD,POLDEGREE,REGULARITY)

【使用说明】该函数计算由 XVAL 和 PSI 给定并用于连续小波变换的小波函数，该小波由向量 YPAT 定义的模式构造，方差为 1。其中模式隐含的 x 值是 xpat=linespace (0,1,length(YPAT))。

常数 NC 的选取应保证通过以下方式的最小二乘拟合，NC*PSI 在区间[0,1]上近似于 YPAT：

- 当 METHOD 等于'polynomial'时，为 POLDEGREE 阶多项式。
- 当 METHOD 等于'othconst' 时，为正交函数空间的投影。

参数 REGULARITY 定义了在 0 和 1 点的边界约束，可以是'continuous'、'differentiable' 或者'none'。

当 METHOD 为'polynomial'时：

- 如果 REGULARITY 等于'continuous'，则必须 POLDEGREE≥3。
- 如果 REGULARITY 等于'differentiable'，则必须 POLDEGREE≥5。

2. cwt 函数

【函数功能】一维连续小波变换函数。

【语法格式】

① COEFS = cwt(S,SCALES,'wname')

② COEFS = cwt(S,SCALES,'wname','plot')

③ COEFS = cwt(S,SCALES,'wname',PLOTMODE)

④ COEFS = cwt(S,SCALES,'wname',PLOTMODE,XLIM)

【使用说明】cwt 为一维小波变换函数。

格式①采用'wname'小波，在正、实尺度 SCALES 下计算向量一维小波系数。S 为被分析信号，小波可以是实数，也可以是虚数。

格式②除了计算小波系数以外，还加以图形显示。

格式③计算并画出连续小波变换的系数，并使用 PLOTMODE 对图形着色。表 10-1 所示是 PLOTMODE 字符串的有效值。

格式④能够计算并画出连续小波变换系数。系数使用 PLOTMODE 和 XLIM 进行着色。其中：XLIM = [x1 x2]并且 1≤x1≤x2≤length(S)。

9.1.4 一维离散小波变换的函数简介

表 9-13 统计了 Matlab R14 提供的一维离散小波变换函数，下面对这些函数进行详细的介绍。

1. dwt 函数

【函数功能】一维单尺度离散小波变换。

【语法格式】

① [cA,cD] = dwt(X,'wname')

② [cA,cD] = dwt(X,'wname','mode',MODE)

③ [cA,cD] = dwt(X,Lo_D,Hi_D)

④ [cA,cD] = dwt(X,Lo_D,Hi_D,'mode',MODE)

【使用说明】dwt 命令使用特定小波（'wname'）或者特定的小波分解滤波器（Lo_D 和 Hi_D）执行单层一维小波分解。

格式①计算低频系数向量 cA 和高频系数向量 cD，由向量 X 进行小波分解得到。字符串'wname'包含小波名。

格式③和上面一样计算小波分解，Lo_D 和 Hi_D 滤波器为输入。其中 Lo_D 是分解低通滤波器，Hi_D 是分解高通滤波器。这两个滤波器必须具有相同的长度。

令 lx 为 X 的长度，lf 为滤波器 Lo_D 和 Hi_D 的长度，如果 DWT 扩展模式设置为 periodization，length(cA) = length(cD) = la，这里 la = ceil(lx/2)。对于其他扩展模式，那么 la = floor(lx+lf-1)/2)。

格式[cA,cD] = dwt(...,'mode',MODE)使用指定的 MODE 扩展模式计算小波分解。

2. wavedec 函数

【函数功能】一维多尺度小波分解（一维多分辨率分析函数）。

【语法格式】

① [C,L] = wavedec(X,N,'wname')

② [C,L] = wavedec(X,N,Lo_D,Hi_D)

【使用说明】wavedec 使用给定的小波'wname'或者小波分解滤波器（Lo_D 和 Hi_D）进行一维多尺度小波分析。

格式①返回信号 X 在 N 层的小波分解。N 必须是严格的正整数。输出分解结构包含小波分解向量 C 和相应的记录向量 L。

格式②使用指定的低通和高通分解滤波器，返回分解结构。

3. wmaxlev 函数

【函数功能】小波分解的最大尺度。

【语法格式】L = wmaxlev(S,'wname')

【使用说明】该函数是一维或者二维小波函数或小波包的导向函数，它返回信号或者图像的最大分解尺度，使用它可以避免分解时超过了这个值。

一般情况下，小波分解尺度都小于理论上的最大值。通常一维分解时分解尺度为 5，而二维分解尺度为 3。

4. idwt 函数

【函数功能】一维单尺度小波逆变换。

【语法格式】

① X = idwt(cA,cD,'wname')

② X = idwt(cA,cD,Lo_R,Hi_R)

③ X = idwt(cA,cD,'wname',L)

④ X = idwt(cA,cD,Lo_R,Hi_R,L)

⑤ X = idwt(...,'mode',MODE)

【使用说明】

idwt 命令使用指定的小波（'wname'）或者小波重构滤波器（Lo_R 和 Hi_R），进行一维单尺度重构。

格式①返回使用小波'wname'，返回单尺度重构的低频系数向量 X 及高频系数向量 cA 和 cD。

格式②使用的是低通滤波器 Lo_R 和重构高通滤波器 Hi_R，这两个滤波器必须有同样的长度。

令 la 为 cA 的长度（即 cD 的长度），lf 为滤波器 Lo_R 和 Hi_R 的长度，如果 DWT 扩展模式设置为 periodization，那么 length(X) = LX，这里 LX = 2×la。对于其他扩展模式，那么 LX = 2*la−lf+2。

格式③和④返回由 idwt(cA,cD, 'wname')得到的长度为 L 的中间部分。L 必须小于 LX。

格式⑤使用指定的延拓模式 MODE 进行小波重构。

格式 X = idwt(cA,[],...)基于低频系数向量 cA，返回单尺度重构低频系数 X。

格式 X = idwt([],cD,...)基于高频系数向量 cD，返回单尺度重构高频系数 X。

5. waverec 函数

【函数功能】一维多尺度小波重构。

【语法格式】

① X = waverec(C,L,'wname')

② X = waverec(C,L,Lo_R,Hi_R)

【使用说明】waverec 使用指定的小波（'wname'）或者重构滤波器（Lo_R and Hi_R）进行一维多尺度小波重构。它返回的是原信号 X，即：

X＝waverec(wavedec(X,N,'wname'),'wname')

因此可以看成是 wavedec 的逆函数。

格式①基于多尺度小波分解结构[C,L]和小波'wname'重构信号 X。

格式②使用指定的重构滤波器重构 X。其中 Lo_R 是重构低通滤波器，Hi_R 为重构高通滤波器。

X = waverec(C,L,'wname')等同于 X = appcoef(C,L,'wname',0)。

6. wrcoef 函数

【函数功能】对一维小波系数进行单支重构。

【语法格式】

① X = wrcoef('type',C,L,'wname',N)

② X = wrcoef('type',C,L,Lo_R,Hi_R,N)

③ X = wrcoef('type',C,L,'wname')

④ X = wrcoef('type',C,L,Lo_R,Hi_R)

【使用说明】wrcoef 基于小波分解结构[C,L]，以及指定的小波'wname'或者重构滤波器 Lo_R 和 Hi_R，来重构一维信号的系数。

格式①基于小波分解结构[C,L]在 N 层计算重构系数向量，'wname'是包含小波名的字符串。

变量'type'决定重构的系数是低频('type' = 'a')还是高频('type' = 'd')。当'type'='a'时，N 可以是 0，否则 N 必须是严格的正整数，且 N≤length(L)−2。

格式②根据指定的重构滤波器计算系数。

格式③和④重构系数的最大层数为 N = length(L)−2。

7. upcoef 函数

【函数功能】一维小波系数的直接重构。

【语法格式】

① Y = upcoef(O,X,'wname',N)

② Y = upcoef(O,X,'wname',N,L)

③ Y = upcoef(O,X,Lo_R,Hi_R,N)

④ Y = upcoef(O,X,Lo_R,Hi_R,N,L)

⑤ Y = upcoef(O,X,'wname')

⑥ Y = upcoef(O,X,Lo_R,Hi_R)

【使用说明】该函数是一维小波分析的函数。

格式①计算向量 X 向上 N 步的重构系数。'wname'是包含小波名的字符串。N 必须是严格的正整数。

如果O='a'，则重构低频系数。如果O='d'，则重构高频系数。

格式②计算向量X向上N步的重构系数，并取出结果中长度为L的中间部分。

如果不给定小波名，可以使用滤波器。格式③和④中，Lo_R是重构低通滤波器，Hi_R是重构高通滤波器。

Y = upcoef(O,X,'wname') 等同于 Y = upcoef(O,X,'wname',1)。

Y = upcoef(O,X,Lo_R,Hi_R) 等同于 Y = upcoef(O,X,Lo_R,Hi_R,1)。

8. detcoef 函数

【函数功能】提取一维小波细节系数。

【语法格式】

① D = detcoef(C,L,N)

② D = detcoef(C,L)

【使用说明】

detcoef是一维小波分析函数。

格式①由小波分解结构[C,L]提取N层细节系数。

N必须是正整数，并且1≤N≤NMAX，NMAX = length(L)–2。

格式②提取最后一层NMAX的细节系数。如果N是由正整数组成的向量，并且1≤N(j)≤NMAX，DCELL = detcoef(C,L,N,'cells')返回一个单元阵列，DCELL{j}包含细节N(j)的系数。DCELL = detcoef(C,L,'cells')等同于DCELL = detcoef(C,L,[1:NMAX])。[D1, ... ,Dp] = detcoef(C,L,[N(1), ... ,N(p)])提取层[N(1), ... ,N(p)]的细节系数。

9. appcoef 函数

【函数功能】提取一维近似系数。

【语法格式】

① A = appcoef(C,L,'wname',N)

② A = appcoef(C,L,'wname')

③ A = appcoef(C,L,Lo_R,Hi_R)

④ A = appcoef(C,L,Lo_R,Hi_R,N)

【使用说明】appcoef是一维小波分析函数，计算一维信号系数的近似值。

格式①使用小波分解框架[C,L]计算N层系数的近似值（参见wavedec函数）。'wname'是包含小波名的字符串。层数N必须是正整数，并且0≤N≤length(L)–2。

格式②提取最大层，即length(L)–2层的系数近似值。

除了直接给定小波名以外，还可以通过给定滤波器的方法。格式③、④中，Lo_R是重构低通滤波器，Hi_R是重构高通滤波器。

10. upwlev 函数

【函数功能】一维单尺度小波分解。

【语法格式】

① [NC,NL,cA] = upwlev(C,L,'wname')

② [NC,NL,cA] = upwlev(C,L,Lo_R,Hi_R)

【使用说明】upwlev 时单尺度小波分析函数。

格式①用于对小波分解结构[C,L]进行单尺度重构，返回新的结构[NC,NL]并提取最后一尺度的低频系数向量 cA。

如果[C,L]是 n = length(L)-2 尺度的分解结构，那么[NC,NL]是尺度 n-1 的分解结构，cA 是尺度 n 的低频系数向量。

'wname'是包含小波名的字符串，C 是原始小波分解向量，L 是相应的记录向量。

除了给定小波名，还可以给定滤波器。对于格式②，Lo_R 是重构低通滤波器，Hi_R 是重构高通滤波器。

11. dwtmode 函数

【函数功能】离散小波变换扩展模式。

【语法格式】

① ST = dwtmode

② dwtmode('mode')

【使用说明】dwtmode 命令用来设置信号或者图像进行离散小波和小波包变换的扩展模式。扩展模式表示信号或者图像分析时边界问题的处理方法。

dwtmode 或者 dwtmode('status')显示当前模式。

ST = dwtmode 或者 ST = dwtmode('status') 显示并返回当前模式，由变量 ST 表示。

ST = dwtmode('status','nodisp')返回当前模式，由变量 ST 表示，但 MATLAB 命令窗口中无显示信息。

'mode'为 DWT 扩展模式，它的类型如表 9.13 所示。

表 9.13　　DWT 扩展模式

'mode'	DWT 扩展模式
'sym'或'symh'	对称延拓模式（half-point）：对称复制边缘值。这是默认模式
'symw'	对称延拓模式（whole-point）：对称复制边缘值
'asym'或'asymh'	反对称延拓模式（half-point）：反对称复制边缘值
'asymw'	反对称延拓模式（whole-point）：反对称复制边缘值
'zpd'	补零
'spd'或'sp1'	1 阶平滑模式（边缘处进行 1 阶导数插补）
'sp0'	0 阶平滑模式（边缘处进行常数延拓）
'ppd'	周期延拓模式（边缘处进行周期延拓）

这几种模式对于 DWT 来说，显得有点多余。然而无论 DWT 采用哪种延拓模式，采用 IDWT 都可以完美地重构。

dwtmode('per')设置 DWT 模式为周期延拓。这种模式产生最短长度的小波分解，但是用于 IDWT 的扩展模式一定要一样，以保证重构的效果。

使用这种模式，dwt 和 dwt2 产生与以前版本的函数 dwtper 和 dwtper2 一样的结果。

所有的函数和 GUI 工具包括一维、二维 DWT，以及一维、二维小波包变换使用指定的 DWT 扩展模式。

dwtmode 可以更新全局变量，使用这 6 种信号延拓模式。这些延拓模式必须通过该函数进行改变，避免直接更改全局变量。

默认的模式从当前路径下的文件 DWTMODE.DEF 载入。如果该文件不存在，可使用 DWTMODE.CFG 文件（路径为 toolbox/wavelet/wavelet）。

dwtmode('save',MODE) 将 MODE 作为一种新的缺省模式存储到 DWTMODE.DEF 文件，如果当前目录下该文件已经存在，那么先删除然后再存储。

dwtmode('save') 等同于 dwtmode('save',CURRENTMODE)。

9.1.5 一维平稳小波变换的函数简介

1. swt 函数

【函数功能】一维离散平稳小波变换。

【语法格式】

① SWC = swt(X,N,'wname')

② SWC = swt(X,N,Lo_D,Hi_D)

③ [SWA,SWD] = swt(X,N,'wname')

④ [SWA,SWD] = swt(X,N,Lo_D,Hi_D)

【使用说明】swt 使用指定的正交小波（'wname'）或者分解滤波器（Lo_R 和 Hi_R），进行一维多尺度平稳小波分解。

格式①使用小波'wname'，计算信号 X 的尺度为 N 的平稳小波分解。N 必须是严格的正整数，信号 X 的长度必须是 2^N 的倍数。

格式③使用低通分解滤波器 Lo_R 和高通分解滤波器对 X 进行分解，返回信号 X。Lo_R 和 Hi_R 具有相同的长度。

输出矩阵 SWC 包含行方式存储的系数向量。当 1≤i≤N 时，输出矩阵 SWC(i,:)包含第 i 层的高频系数，SWC(N+1,:)包含第 N 层的低频系数。

格式[SWA,SWD] = swt(...)计算低频系数 SWA 和高频系数 SWD 等平稳小波系数。系数向量以行方式存储的。对于 1≤i≤N，输出矩阵 SWA(i,:)包含第 i 层的低频系数，SWD(i,:)包含第 i 层的高频系数。

2. iswt 函数

【函数功能】一维离散平稳小波逆变换。

【语法格式】

① X = iswt(SWC,'wname')

② X = iswt(SWA,SWD,'wname')

③ X = iswt(SWC,Lo_R,Hi_R)

④ X = iswt(SWA,SWD,Lo_R,Hi_R)

【使用说明】iswt 使用指定的正交小波（'wname'）或者重构滤波器（Lo_R 和 Hi_R），进行一维多尺度平稳小波重构。

格式①、②或 X = iswt(SWA(end,:),SWD, 'wname')基于多尺度平稳小波分解结构 SWC 或 [SWA,SWD]，重构信号 X。

格式③、④或 X = iswt(SWA(end,:),SWD,Lo_R,Hi_R)使用重构低通滤波器 Lo_R 和重构高通滤波器 Hi_R，返回信号 X。Lo_R 和 Hi_R 具有相同的长度。

9.1.6 二维离散小波变换的函数简介

1. dwt2 函数

【函数功能】二维单尺度小波变换。

【语法格式】

① [cA,cH,cV,cD] = dwt2(X,'wname')

② [cA,cH,cV,cD] = dwt2(X,Lo_D,Hi_D)

【使用说明】dwt2 使用指定的小波（'wname'）或者分解滤波器（Lo_D 和 Hi_D）进行二维单尺度小波分解。

格式①根据输入矩阵 X 进行小波分解，计算低频系数矩阵 cA 和高频系数矩阵 cH（水平方向）、cV（垂直方向）、cD（对角方向）。

格式②基于指定的小波分解滤波器计算二维小波分解系数。其中 Lo_D 是分解低频滤波器，Hi_D 是分解高频滤波器。Lo_D 和 Hi_D 必须具有相同的长度。

令 sx = size(X)，*lf* 为滤波器的长度，那么对于 DWT 的周期延拓模式来说，size(cA) = size(cH) = size(cV) = size(cD) = sa，这里 sa = ceil(sx/2)。对于其他延拓模式，sa = floor((sx+*lf*−1)/2)。

函数[cA,cH,cV,cD] = dwt2(...,'mode',MODE)根据指定的延拓模式 MODE 计算小波分解系数。例如：[cA,cH,cV,cD] = dwt2(x,'db1','mode','sym');

2. wavedec2 函数

【函数功能】二维多尺度小波分解（二维多分辨率分析函数）。

【语法格式】

① [C,S] = wavedec2(X,N,'wname')

② [C,S] = wavedec2(X,N,Lo_D,Hi_D)

【使用说明】wavedec2 是二维小波分析函数。格式①使用小波'wname'返回矩阵 X 尺度为 N 时的小波分解。输出是分解向量 C 和相应的记录矩阵 C。N 必须是严格的正整数。

除了给定小波名以外，还可以使用滤波器。格式②中，Lo_D 是分解低通滤波器，Hi_D 是分解高通滤波器。

向量 C 的形式为：

C = [A(N) | H(N) | V(N) | D(N) | ···H(N−1) | V(N−1) | D(N−1) | ··· | H(1) | V(1) | D(1)]

其中 A、H、V、D 是行向量。A 是低频系数，H 是高频水平方向系数，V 是高频垂直方向系数，D 是高频对角方向的系数。每个向量都是以列的形式存储于矩阵 C 中。

矩阵 S 为：

S(1,:)——尺度为 N 的低频系数的长度；

S(i,:)——尺度为 N−i+2 时的高频系数的长度，其中 i = 2, …, N+1；

S(N+2,:) = size(X)。

3. idwt2 函数

【函数功能】二维单尺度小波逆变换。

【语法格式】

① X = idwt2(cA,cH,cV,cD,'wname')

② X = idwt2(cA,cH,cV,cD,Lo_R,Hi_R)

③ X = idwt2(cA,cH,cV,cD,'wname',S)

④ X = idwt2(cA,cH,cV,cD,Lo_R,Hi_R,S)

⑤ X = idwt2(…,'mode',MODE)

【使用说明】idwt2 命令使用给定的小波（'wname'）或者小波重构滤波器（Lo_R 和 Hi_R）进行二维单尺度小波重构。

格式①使用小波'wname'，基于低频矩阵 cA 和高频水平方向矩阵 cH，垂直方向矩阵 cV 和对角方向矩阵 cD，计算单尺度重构低频系数矩阵 X。

格式②使用的是指定滤波器。Lo_R 是重构低通滤波器，Hi_R 是重构高通滤波器。

令 sa = size(cA) = size(cH) = size(cV) = size (cD)，*lf* 为滤波器长度，那么如果 DWT 是周期延拓模式，则 size(X) = SX，其中 X = 2* SA。对于其他延拓模式，SX = 2*size(cA)−*lf*+2。

格式③和④使用语法 idwt2 (cA,cH,cV,cD,'wname') 返回尺度为 S 的中间部分。S 必须小于 SX。

格式 X = idwt2(cA,[],[],[],…)基于低频系数矩阵 cA，返回单尺度重构低频系数矩阵 X。

格式 X = idwt2([],cH,[],[],…)基于水平方向高频系数 cH，返回单尺度重构高频系数。同样，对于垂直方向和高频方向，相应的格式是 X = idwt2([],[],cV,[],…)和 X=idwt2([],[],[],cD,…)。

更一般的情况是，格式 X= idwt2 (AA,HH,VV,DD,…)返回单尺度重构矩阵 X，这里 AA 可以是 cA 或者[]等。

4. waverec2 函数

【函数功能】二维多尺度小波重构。

【语法格式】

① X = waverec2 (C,S,'wname')

② X = waverec2 (C,S,Lo_R,Hi_R)

【使用说明】waverec2 是二维小波分析函数，可以认为它是 wavedec2 的逆函数。格式①基于小波分解结果[C,S]对矩阵 X 进行多尺度小波重构。'wname'是包含小波名的字符串。

除了给定小波以外，还可以使用滤波器。对于格式②，Lo_R 是重构低通滤波器，Hi_R 是重构高通滤波器。

X = waverec2 (C,S,'wname')等同于 X = appcoef2 (C,S,'wname',0)。

5. wrcoef2 函数

【函数功能】对二维小波系数进行单支重构。

【语法格式】

① X = wrcoef2 ('type',C,S,'wname',N)

② X = wrcoef2 ('type',C,S,Lo_R,Hi_R,N)

③ X = wrcoef2 ('type',C,S,'wname')

④ X = wrcoef2 ('type',C,S,Lo_R,Hi_R)

【使用说明】wrcoef2 是二维小波分析函数，用来重构一幅图像的系数。格式①基于小波分解结构，计算尺度为 N 时的重构系数矩阵。

'wname'是包含小波名的字符串。如果'type' = 'a'，则重构低频系数；'type' = 'h'、'v' 、'd'时，分别重构高频的水平、垂直和对角系数。

N 必须是整数，如果'type' = 'a'，0≤N≤size(S,1)−2。如果'type' = 'h'，'v'或者'd'，1≤N≤size(S,1)−2。

除了给定小波名以外，还可以使用滤波器。对于格式②，Lo_R 是重构低通滤波器，Hi_R 是重构高通滤波器。

格式③和④重构最大尺度 N=size (S,1) −2 的系数。

6. upcoef2 函数

【函数功能】二维小波系数的直接重构。

【语法格式】

① Y = upcoef2 (O,X,'wname',N,S)

② Y = upcoef2 (O,X,Lo_R,Hi_R,N,S)

③ Y = upcoef2 (O,X,'wname',N)

④ Y = upcoef2 (O,X,Lo_R,Hi_R,N)

⑤ Y = upcoef2 (O,X,'wname')

⑥ Y = upcoef2 (O,X,Lo_R,Hi_R)

【使用说明】upcoef2 是二维小波分析函数。格式①计算矩阵 X 的 N 层重构系数，并提取长度为 S 的中间部分。'wname'是包含小波名的字符串。

O = 'a'，表示重构低频系数；O = 'h'，表示重构水平方向高频系数；O = 'v'，表示重构垂直方向高频系数；O = 'd'，表示重构对角方向高频系数。

除了指定小波名进行重构以外，还可以使用滤波器。对于格式②，Lo_R 是重构低通滤波器，Hi_R 是重构高通滤波器。

格式③和④返回无任何截断的计算结果。

Y = upcoef2 (O,X,'wname')等同于 Y = upcoef2 (O,X,'wname',1)。

Y = upcoef2 (O,X,Lo_R,Hi_R)等同于 Y = upcoef2 (O,X,Lo_R,Hi_R,1)。

7. detcoef2 函数

【函数功能】提取二维小波分解高频系数。

【语法格式】D = detcoef2 (O,C,S,N)

【使用说明】detcoef2 是二维小波分析函数。该函数由小波分解结构[C,S]，提取尺度为 N 时的水平、垂直或者对角高频系数，对应的变量 O 分别为'h'、'v'和'd'。

N必须是整数，且1≤N≤size (S,1)−2。

函数[H,V,D] = detcoef2('all',C,S,N)返回尺度为N时的水平高频系数H，垂直高频系数V及对角高频系数D。

函数D = detcoef2 ('compact',C,S,N) 返回尺度为N时的高频系数，以行方式存储。

detcoef2('a',C,S,N) 等同于 detcoef2 ('all',C,S,N)。

detcoef2('c',C,S,N) 等同于 detcoef2 ('compact',C,S,N)。

8. appcoef2 函数

【函数功能】提取二维小波分解低频系数。

【语法格式】

① A = appcoef2 (C,S,'wname',N)

② A = appcoef2 (C,S,'wname')

③ A = appcoef2 (C,S,Lo_R,Hi_R)

④ A = appcoef2 (C,S,Lo_R,Hi_R,N)

【使用说明】appcoef2 是二维小波分析函数，它计算二维信号的低频系数。

格式①使用小波分级结构[C,S]，计算N层的低频系数。'wname'是包含小波名的字符串；N必须是整数，且0≤N≤size (S,1)−2。

格式②提取最后一层 size (S,1)−2 的低频系数。

除了给定小波以外，还可以给定滤波器。格式③、④中，Lo_R是重构低频滤波器，Hi_R是重构高频滤波器。

9. upwlev2 函数

【函数功能】二维小波分解的单尺度重构。

【语法格式】

① [NC,NS,cA] = upwlev2(C,S,'wname')

② [NC,NS,cA] = upwlev2(C,S,Lo_R,Hi_R)

【使用说明】upwlev2 是二维小波分解函数。格式①对小波分解结构[C,S]进行单尺度重构，返回新的分解结构[NC,NS]，并提取最后一尺度的低频系数矩阵cA。[C,S]是尺度 n = size(S,1) −2 的分解结构，因此[NC,NS]是尺度n−1的分解结构，cA是相应的记录矩阵。

除了对给定小波进行重构以外，还可以使用滤波器。格式②中，Lo_R 是重构低通滤波器，而Hi_R是重构高通滤波器。

10. wenergy2 函数

【函数功能】计算二维小波分解能量。

【语法格式】

① [Ea,Eh,Ev,Ed] = wenergy2(C,S)

② [Ea,EDetail] = wenergy2(C,S)

【使用说明】对于二维小波分解结构[C,S]，格式①返回的Ea是低频部分能量的百分比；Eh、Ev和Ed分别是高频部分水平、垂直和对角方向能量百分比的向量。

格式②返回 Ea 和 Edetail，后者是向量 Eh、Ev 及 Ed 的和。

9.1.7 二维平稳小波变换的函数简介

1. swt2 函数

【函数功能】二维离散平稳小波变换。

【语法格式】

① SWC = swt2(X,N,'wname')

② [A,H,V,D] = swt2(X,N,'wname')

③ SWC = swt2(X,N,Lo_D,Hi_D)

④ [A,H,V,D] = swt2(X,N,Lo_D,Hi_D)

【使用说明】swt2 使用指定的正交小波（'wname'）或者正交小波分解滤波器，进行二维多尺度平稳小波分解。

格式①和②使用小波'wname'，计算信号 X 的尺度为 N 的平稳小波分解。N 必须是严格的正整数，size(X,1)和 size(X,2)必须是 2^N 的倍数。

输出[A,H,V,D]是三维矩阵，其中 A(:,:,*i*)包含第 *i* 层的低频系数，H(:,:,*i*)、V(:,:,*i*)和 D(:,:,*i*)分别包含第 *i* 层的水平方向、垂直方向和对角方向的高频系数。这里 1≤*i*≤N。SWC = [H(:,:,1:N) ; V(:,:,1:N) ; D(:,:,1:N) ; A(:,:,N)]。

格式③和④和以上两个函数一样，计算平稳小波分解。其中，Lo_D 是低通分解滤波器，Hi_D 是高通分解滤波器。Lo_D 和 Hi_D 具有相同的长度。

格式③使用重构低通滤波器 Lo_R 和重构高通滤波器 Hi_R 对 X 进行分解，返回信号 X。Lo_R 和 Hi_R 具有相同的长度。

输出矩阵 SWC 包含行方式存储的系数向量。当 1≤*i*≤N 时，输出矩阵 SWC(*i*,:)包含第 *i* 层的高频系数，SWC(N+1,:)包含第 N 层的低频系数。

格式[SWA,SWD] = swt(…)计算低频系数 SWA 和高频系数 SWD 等平稳小波系数。系数向量以行方式存储。对于 1≤*i*≤N，输出矩阵 SWA(*i*,:)包含第 *i* 层的低频系数，SWD(*i*,:)包含第 *i* 层的高频系数。

2. iswt2 函数

【函数功能】二维离散平稳小波逆变换。

【语法格式】

① X = iswt2(SWC,'wname')

② X = iswt2(A,H,V,D,'wname')

③ X = iswt2(SWC,Lo_R,Hi_R)

④ X = iswt2(A,H,V,D,Lo_R,Hi_R)

【使用说明】iswt2 使用指定的正交小波（'wname'）或者重构滤波器（Lo_R 和 Hi_R），进行二维多尺度平稳小波重构。

格式①、②或 X = iswt2(A(:,:,end),H,V,D, 'wname') 基于多尺度平稳小波分解结构 SWC 或[A,H,V,D]，重构信号 X。

格式③、④或 X = iswt2(A(:,:,end),H,V,D,Lo_R,Hi_R)使用重构低通滤波器 Lo_R 和重构高通滤波器 Hi_R，返回信号 X。Lo_R 和 Hi_R 具有相同的长度。

9.1.8 小波包分析的函数简介

1. wpcoef 函数

【函数功能】小波包系数。

【语法格式】

① X = wpcoef (T,N)

② X = wpcoef (T)

【使用说明】wpcoef 是一维或二维小波包分析函数。

格式①返回小波包树 T 中结点 N 相关的系数。如果 N 不存在，那么 X = [ ]。

格式②等同于 X = wpcoef (T,0)。

2. wpdec 函数

【函数功能】一维小波包分解。

【语法格式】

① T = wpdec (X,N,'wname',E,P)

② T = wpdec (X,N,'wname')

【使用说明】wpdec 是一维小波包分解函数。

格式①使用小波'wname'，并根据向量 X 在第 N 层的小波包分解结构，返回小波包树 T。E 是包含熵类型的字符串，P 是由 T 值决定的优化参数。它们之间的对应关系参见 wentropy 函数的说明。

格式②等价于 T = wpdec (X,N, 'wname', 'shannon')。

E 是包含熵类型的字符串，P 是由 E 值确定的优化参数，如表 9.14 所示。

表 9.14 熵类型 E 和 P 的对应表

熵类型名（E）	参数（P）	说 明
'shannon'		P 不用
'log energy'		P 不用
'threshold'	0≤P	P 是阈值
'sure'	0≤P	P 是阈值
'norm'	1≤P	P 是指数
'user'	string	P 是包含 M 文件名的字符串，M 文件定义了使用单输入 X 的自定义熵函数
FunName	P 无限制	FunName 是除了前面的熵类型名以外的任何其他字符串，包含用户熵函数的 M 文件名，该熵函数以 X 作为输入，P 作为熵函数的附加参数

对于给定的正交小波函数，可以生成一组小波包基，每个基都提供了一种特定的信号编

码方法，它能保留信号的全部能量，并对信号的特征进行准确的重构。这些小波包可以用于对给定信号进行多种分析和解释。

对于小波包的分解和最佳分解方式的选择，都有简单而有效的算法。其中自适应滤波器算法可以直接用于最优信号编码和数据压缩。

在正交小波分解过程中，一般的方法是将低频系数分为两部分，得到新的低频系数向量和高频系数向量，两个连续的低频系数之间损失的信息可以由高频系数获得，然后是将新的低频系数向量继续分解，而高频系数保持不动。而对于小波包分解来说，每个高频系数向量也使用和低频系数分解同样的方法，分成两个部分，这样就提供了更为丰富的信号分析方法：在一维分析中产生了完整二叉树，而在二维分析中则产生四叉树。

3. wpdec2 函数

【函数功能】二维小波包分解。

【语法格式】

① T = wpdec2(X,N,'wname',E,P)

② T = wpdec2(X,N,'wname')

【使用说明】wpdec2 是二维小波包分解函数。

格式①使用特定的小波'wname'，由矩阵 X 在第 N 层的小波包分解，返回小波包树 T。

格式②等同于 T = wpdec2(X,N, 'wname', 'shannon')。

E 是包含熵类型的字符串，P 是由 E 值确定的优化参数。

4. wpsplt 函数

【函数功能】分割（分解）小波包。

【语法格式】

① T = wpsplt(T,N)

② [T,cA,cD] = wpsplt(T,N)

③ [T,cA,cH,cV,cD] = wpsplt(T,N)

【使用说明】wprec 是一维或二维小波包分析函数，它在对结点分解后更新小波包树。

格式①根据结点 N 的分解，返回修改后的小波包树 T。

对于一维分解，使用格式②。其中 cA 是结点 N 的低频系数，cD 是结点 N 的高频系数。

对于二维分解，使用格式③。其中 cA 是结点 N 的低频系数，cH、cV、cD 分别是水平、垂直和对角方向的高频系数。

5. wprcoef 函数

【函数功能】重构小波包系数。

【语法格式】X = wprcoef (T,N)

【使用说明】wprcoef 是一维或二维小波包分析函数。该函数计算小波包树 T 中结点 N 的重构系数。

X = wprcoef (T) 等同于 X = wprcoef (T,0)。

6. wprec 函数

【函数功能】一维小波包重构。

【语法格式】X = wprec (T)

【使用说明】wprec 是一维小波包分析函数，它根据小波包树 T，返回重构的向量 X。wprec 是 wpdec 的逆函数，即 X＝wprec(wpdec (X,'wname'))。

7. wprec2 函数

【函数功能】二维小波包重构。

【语法格式】X = wprec2 (T)

【使用说明】wprec 是二维小波包分析函数，它根据小波包树 T，返回重构的向量 X。wprec2 是 wpdec2 的逆函数，即 X＝wprec2 (wpdec2 (X,'wname'))。

8. wpjoin 函数

【函数功能】重新组合小波包。

【语法格式】

① T = wpjoin (T,N)

② [T,X] = wpjoin (T,N)

③ T = wpjoin (T)

④ [T,X] = wpjoin (T)

【使用说明】wpjoin 是一维或二维小波包分析函数，它在重新组合结点后更新小波包树。这些结点从左到右，从上到下进行编号，根结点序号是 0。

格式①返回修改后的结点 N 重新组合后的小波包树 T。格式②也返回结点的系数。格式③等同于 T = wpjoin (T,0)。格式④等同于[T, X] = wpjoin(T,0)。

9. besttree 函数

【函数功能】最优小波包树。

【语法格式】

① T = besttree (T)

② [T,E] = besttree (T)

③ [T,E,N] = besttree (T)

【使用说明】besttree 是一维或二维小波包分析函数，它根据某种熵标准计算原始树的最佳子树，得到的结果可能比原始树小得多。

长度为 $N = 2^L$ 的信号可以用 2^L 种不同的分解方式分解，这正好是一个深度为 L 的完整二叉树的二叉子树的数目，是非常庞大的，如果用枚举法一一列举将难以想象。因此，我们感兴趣的是要根据某种简单的准则和有效的算法，找出其最优分解方式。

格式①根据最优熵值，计算相应的最优树 T。格式②计算最优树 T 和最优熵值 E。索引序号为 j-1 的结点的最优熵值为 E(j)。格式③计算最优树 T，最优熵 E 和包含被合并结点的

索引序号向量 N。

10. bestlevt 函数

【函数功能】最优完整小波包树。

【语法格式】

① T = bestlevt (T)

② [T,E] = bestlevt (T)

【使用说明】bestlevt 是一维或二维小波包分析函数，它根据某种熵标准计算出初始树的最佳完整子树，结果得到的完整树可能比原始树的深度小一些。该函数在功能上和 besttree 差不多，但它计算的结果是完整子树，而后者就不一定。

格式①根据最优深度小波包树的分解，计算修改的小波包树 T。格式②除了计算最优小波包树 T 以外，还得到最优熵值 E。索引号为 $j-1$ 的结点的最优熵值为 $E(j)$。

11. entrupd 函数

【函数功能】更新小波包的熵值。

【语法格式】

① T = entrupd (T,ENT)

② T = entrupd (T,ENT,PAR)

【使用说明】entrupd 是一维或二维小波包工具。格式①或②对于给定的小波包树 T，使用熵函数 ENT 和优化参数 PAR，返回更新树 T。

12. wentropy 函数

【函数功能】计算小波包的熵值。

【语法格式】

① E = wentropy (X,T,P)

② E = wentropy (X,T)

【使用说明】格式①返回向量或矩阵 X 的熵值 E。格式②等同于 E = wentropy(X,T,0)。对于这两种输入，输出 E 都是实数。

E 是包含熵类型的字符串，P 是由 E 值确定的优化参数。

13. wp2wtree 函数

【函数功能】从小波包树中提取小波树。

【语法格式】T = wp2wtree (T)

【使用说明】wp2wtree 是一维或二维小波包分析函数。该函数根据小波分解树，计算修改后的小波包树 T。

14. wpcutree 函数

【函数功能】截除小波包分解树。

【语法格式】

① T = wpcutree(T,L)

② [T,RN] = wpcutree(T,L)

【使用说明】wpcutree 是一维或二维小波包分析函数。

格式①截除树 T 中 L 层以下的所有子树。格式②中，向量 RN 包含重构结点的索引号。

15. wpfun 函数

【函数功能】小波包函数。

【语法格式】

① [WPWS,X] = wpfun('wname',NUM,PREC)

② [WPWS,X] = wpfun('wname',NUM)

【使用说明】wpfun 是小波包分析函数。

格式①计算小波'wname'的小波包，其时间长度为 2^{PREC}。PREC 必须是正整数。输出矩阵 WPWS 包含下标从 0 到 NUM 的 W 函数 $[W_0; W_1; \dots ; W_{NUM}]$ ，并按行向量存储。输出向量 X 是相应的普通 X-网格矢量。格式②等同于[WPWS,X] = wpfun('wname',NUM,7)。

9.1.9　小波降噪和压缩函数简介

1. wnoise 函数

【函数功能】产生小波的噪声测试数据。

【语法格式】

① X = wnoise(FUN,N)

② [X,XN] = wnoise(FUN,N,SQRT_SNR)

③ [X,XN] = wnoise(FUN,N,SQRT_SNR,INIT)

【使用说明】格式①返回由 FUN 给定的区间[0,1]上的测试信号，共有 2^N 点。格式②返回测试向量 X，std(X) = SQRT_SNR。另外一个返回的向量 XN 是受高斯白噪声 N(0,1)污染的测试向量，XN 的信噪比 SNR = $(SQRT_SNR)^2$。格式③同样返回 X 和 XN，其种子为 INIT。参数 FUN 的含义如下：FUN = 1——'blocks'；FUN = 2——'bumps'；FUN = 3——'heavy sine'；FUN = 4——'doppler'；FUN = 5——'quadchirp'；FUN = 6——'mishmash。

2. wnoisest 函数

【函数功能】估计一维小波系数的标准差。

【语法格式】STDC = wnoisest (C,L,S)

【使用说明】该函数返回输入向量 S 中各层高频系数的标准差。[C,L]是输入的小波分解结构。

如果 C 是一维数组，STDC = wnoisest(C)返回一个向量，STDC(k)是 C{k}标准差的估计。

估计是由中值绝对偏差除以 0.6745 得到标准差，非常适合于用一维零均值高斯白噪声降

噪的信号模型。

3. ddencmp 函数

【函数功能】获取降噪或压缩的默认值。

【语法格式】

① [THR,SORH,KEEPAPP,CRIT] = ddencmp(IN1,IN2,X)

② [THR,SORH,KEEPAPP] = ddencmp(IN1,'wv',X)

③ [THR,SORH,KEEPAPP,CRIT] = ddencmp(IN1,'wp',X)

【使用说明】ddencmp 是降噪和压缩的导向函数。它给出一维或二维信号使用小波或小波包进行降噪和压缩一般过程的所有默认值。

格式①对于输入向量或矩阵 X，使用小波或小波包返回降噪或压缩的默认值。X 可以是一维或二维信号。THR 是阈值，SORH 是软阈值或硬阈值。KEEPAPP 允许用户保存低频系数。CRIT（只用于小波包）是熵名称。

IN1 为'den'时，表示降噪；为'cmp'时，表示压缩。

IN2 为'wv'时，表示小波；IN2 为'wp'时，表示小波包。

对于小波而言（3 个输出变量）：格式②返回降噪的默认值（IN1 为'den'），或 X 的压缩（IN1 为'cmp'）。这些值可以用于 wdencmp 函数。

对于小波包而言（4 个输出变量）：格式③返回降噪的默认值（IN1 为'den'），或 X 的压缩（IN1 为'cmp'）。这些值可以用于 wpdencmp 函数。

4. wthresh 函数

【函数功能】进行软阈值或硬阈值处理。

【语法格式】Y = wthresh(X,SORH,T)

【使用说明】该函数返回输入向量或矩阵 X 进行阈值处理的结果。SORH = 's'时，进行软阈值处理，即把信号的绝对值与指定的阈值进行比较，小于或等于阈值的点变为 0；大于阈值的点变为该点值与阈值的差。而 SORH = 'h'时，则进行硬阈值处理，即把信号的绝对值与指定的阈值进行比较，小于或等于阈值的点变为 0；大于阈值的点保持不变。T 是阈值大小。一般来说，硬阈值比软阈值处理后的信号更加粗糙。

5. thselect 函数

【函数功能】选择降噪时的阈值。

【语法格式】THR = thselect (X,TPTR)

【使用说明】thselect 是一维降噪的导向函数。

该函数由字符串 TPTR 定义的选择规则，返回阈值 X。

根据基本的噪声模型 $y=f(t)+e$（其中 e 是白噪声 $N(0,1)$），选取阈值时有如下 4 个规则，其中每一条规则对应于函数 thselect 中输入参数 tptr 的一个选项。

6. wbmpen 函数

【函数功能】设置一维或二维小波降噪的阈值。

【语法格式】THR = wbmpen(C,L,SIGMA,ALPHA)

【使用说明】该函数返回降噪时的全局阈值 THR。THR 使用 Birge-Massart 惩罚函数方法，由小波系数选择规则得到。[C,L]是信号或图像降噪时的小波分解结构。SIGMA 是降噪模型中零均值高斯白噪声的标准差。ALPHA 是调整参数。它必须是大于 1 的实数。其值越大，降噪信号或图像的小波表示越稀疏。ALPHA 的典型值为 2。

THR 由极小化下式的惩罚规则得到：

crit(t) = -sum(c(k)^2,k≤t) + 2*SIGMA^2*t*(ALPHA + log(n/t))

这里 c(k)是小波系数，它是按绝对值递减的顺序存储的。n 是小波系数的个数。设 t^*是上式的极小值，那么 THR＝$|c(t^*)|$。

wbmpen(C,L,SIGMA,ALPHA,ARG)计算全局阈值，并且还画出 3 个曲线图。

7. wdcbm 函数

【函数功能】使用 Birge-Massart 策略的一维小波阈值。

【语法格式】

① [THR,NKEEP] = wdcbm(C,L,ALPHA)

② [THR,NKEEP] = wdcbm(C,L,ALPHA,M)

【使用说明】

格式①对于降噪或压缩，返回尺度相关的阈值 THR 和系数的个数 NKEEP。THR 使用基于 Birge-Massart 策略的小波系数选择规则得到。

[C,L]是待降噪或压缩信号在 j = length(L) −2 层的小波分解结构。ALPHA 和 M 必须是大于 1 的实数。THR 是长度为 j 的向量，THR(i)包含了第 i 层的阈值。NKEEP 是长度 j 的向量，NKEEP(i)包含第 i 层的系数个数。

j、M 和 ALPHA 定义了 Birge-Massart 策略：

- 在第 j+1 层（或近似层），所有的都保存。
- 对于从 1 到 j 的第 i 层，较大的 n_i 系数被保存，其中 $n_i = M\,(j+2-i)^{ALPHA}$，典型的 ALPHA 值为 1.5，降噪时，ALPHA = 3。

由于前面的规则推出 i 的值是从 j+1 到 n_{j+1} = M = L(1)，因此 M 的默认值是 M = L(1)，推荐 M 的值从 L(1)到 2*L(1)。L(1)是低频系数的个数。

格式 wdcbm(C,L,ALPHA)等同于 wdcbm(C,L,ALPHA,L(1))。

8. wdcbm2 函数

【函数功能】使用 Birge-Massart 策略的二维小波阈值。

【语法格式】

① [THR,NKEEP] = wdcbm2(C,S,ALPHA)

② [THR,NKEEP] = wdcbm2(C,S,ALPHA,M)

【使用说明】格式②对于降噪或压缩，返回尺度相关的阈值 THR 和系数的个数 NKEEP。THR 使用基于 Birge-Massart 策略的小波系数选择规则得到。[C,S]是待降噪或压缩图像在 j = size(S,1) −2 层的小波分解结构。其他参数的含义如同 wdcbm 函数。

9. wthcoef 函数

【函数功能】一维小波系数阈值处理。

【语法格式】

① NC = wthcoef ('d',C,L,N,P)

② NC = wthcoef ('d',C,L,N)

③ NC = wthcoef ('a',C,L)

④ NC = wthcoef ('t',C,L,N,T,SORH)

【使用说明】wthcoef 是一维小波降噪和压缩的导向函数。

格式①返回由小波分解结构[C,L]，并根据向量 N 和 P 定义的压缩比得到的系数。N 包含被压缩的高频尺度，P 是一个包含把较小系数置零的百分比信息的向量。N 和 P 必须具有相同的长度。向量 N 必须满足：$1 \leqslant N(i) \leqslant length(L)-2$。

格式②返回由[C,L]将 N 中定义的高频尺度的系数置零后的系数。

格式③返回将低频系数置零后的系数。

格式④返回由小波分解结构[C,L]通过软阈值（SORH='s'）或硬阈值（SORH='h'）处理后的系数。N 包含要经过阈值处理的高频尺度，T 是对应的阈值。N 和 T 必须有相同的长度。

[NC,L]是修改后的小波分解结构。

10. wthcoef2 函数

【函数功能】二维小波系数阈值处理。

【语法格式】

① NC = wthcoef2('type',C,S,N,T,SORH)

② NC = wthcoef2('type',C,S,N)

③ NC = wthcoef2('a',C,S)

④ NC = wthcoef2('t',C,S,N,T,SORH)

【使用说明】wthcoef 是二维小波降噪和压缩的导向函数。

对于'type' = 'h'（'v'或'd'），格式①由小波分解结构，并通过软阈值（SORH='s'）或硬阈值（SORH='s'）处理，得到的水平（垂直或对角）方向的系数。N 包含要经过阈值处理的高频尺度，T 是对应的阈值。N 和 T 必须有相同的长度。向量 N 必须满足：$1 \leqslant N(i) \leqslant size(S,1)-2$。

对于'type' = 'h'（'v'或'd'），格式②返回由[C,S]将 N 中定义的高频尺度的系数置零后得到的水平（垂直或对角）方向的系数。

格式③返回将 N 中定义的低频尺度置零后的系数。

格式④返回由小波分解结构[C,S]通过软阈值（SORH='s'）或硬阈值（SORH='h'）处理后的高频系数。N 包含要经过阈值处理的高频尺度，T 是相应的用于 3 个高频方向的阈值。N 和 T 必须有相同的长度。

[NC,S]是修改后的小波分解结构。

11. wden 函数

【函数功能】使用小波进行一维自动降噪。

【语法格式】

① [XD,CXD,LXD] = wden(X,TPTR,SORH,SCAL,N,'wname')

② [XD,CXD,LXD] = wden(C,L,TPTR,SORH,SCAL,N,'wname')

【使用说明】wden 是一维降噪函数，它使用小波对一维信号执行自动降噪的过程。

格式①通过对小波系数阈值比较后，返回输入信号 X 降噪后的信号 XD。附加的输出变量[CXD,LXD]是降噪信号 XD 的小波分解结构。

字符串 TPTR 包含了阈值选择规则，请参见 thselect 函数的使用说明。

SCAL 定义了阈值的调整：

- SCAL='one'时，表示不要调整。
- SCAL='sln'时，表示对第一层系数的层噪声进行一次估计、调整。
- SCAL='mln'时，表示使用对各层噪声分别进行估计、调整。

小波分解在第 N 层，使用的正交小波名是'wname'。

格式②使用和上面相同的选项，并返回相同的变量，但它是由输入信号的小波分解结构[C,L]直接得到的。

对于阈值的选择，可以参阅 thselect 函数。这里需要指出的是：在实际中，不能直接得到信号的基本模型，必须考虑模型的偏差。SCAL 变量即是反映这种模型偏差的变量，它的取值指定了阈值修改方法，如上面所述。

12. wdencmp 函数

【函数功能】使用小波进行降噪和压缩。

【语法格式】

① [XC,CXC,LXC,PERF0,PERFL2] = wdencmp('gbl',X,'wname',N,THR,SORH,KEEPAPP)

② [XC,CXC,LXC,PERF0,PERFL2] = wdencmp('lvd',X,'wname',N,THR,SORH)

③ [XC,CXC,LXC,PERF0,PERFL2] = wdencmp('lvd',C,L,'wname',N,THR,SORH)

【使用说明】该函数是一维或二维小波压缩或降噪的导向函数，它使用小波，对信号或图像执行降噪或压缩过程。

格式①对于输入信号 X（一维或二维），使用全局正阈值 THR，由小波系数阈值得到降噪或压缩后的 XC。

附加的输出变量[CXC,LXC]是 XC 的小波分解结构。PERF0 和 PERFL2 是恢复和压缩的 L2 范数百分比。

如果[C,L]表示 X 的小波分解结构，则

$$\text{PERFL2} = 100 * (\text{vector-norm of CXC / vector-norm of C})^2$$

如果 X 是一维信号，'wname'是正交小波，PERFL2 降为 $\dfrac{100\|XC\|^2}{\|X\|^2}$。

在第 N 层，使用小波'wname'执行小波分解。SORH ('s' or 'h')是软阈值或硬阈值。如果

KEEPAPP = 1，低频系数不能进行阈值处理，否则可以。

格式 wdencmp('gbl',C,L,'wname',N,THR,SORH,KEEPAPP)使用和上面相同的选项，具有同样的输出，但它是由被降噪或压缩信号在第 N 层，使用小波'wname'的输入小波分解结构[C,L]直接得到的。

对于一维和'lvd'情况，格式②和③同样使用和上面相同的选项，具有相同的输出变量。但它允许各层分别设置向量 THR 定义的阈值（THR 长度必须为 N）。另外，它保存低频信号。相对于 wden（自动降噪），wdencmp 函数可以使用自己的降噪策略，具有较大的灵活性。

对于二维和'lvd'情况，格式②或③中，THR 必须是 3×N 矩阵，包含各尺度中水平、垂直和对角 3 个方向的阈值。

对于给定小波基进行压缩，意味着信号在小波域的表示相对缺少了一些信息。但是，规则的信号可以由小部分的低频系数（在合适的小波层上）和一部分高频系数进行重构。

13. wthrmngr 函数

【函数功能】阈值设置管理。

【语法格式】THR = wthrmngr(OPTION,METHOD,VARARGIN)

【使用说明】该函数使用 OPTION 选项返回一个全局阈值或各层不同的阈值。输入 VARARGIN 依赖于 OPTION 和 METHOD 的取值。该 M 文件返回的阈值，可用于整个 Matlab 小波工具箱中的降噪和压缩工具。

9.1.10 小波包降噪和压缩函数简介

1. wpbmpen 函数

【函数功能】小波包降噪的 Penalized 阈值。

【语法格式】

① THR = wpbmpen (T,SIGMA,ALPHA)

② THR = wpbmpen (T,SIGMA,ALPHA,ARG)

【使用说明】该函数返回降噪时的全局阈值 THR。THR 使用 Birge-Massart 惩罚函数方法，由小波包系数选择规则得到。

T 是被降噪信号或图像的小波包分解对应的小波包树。SIGMA 是降噪模型中零均值高斯白噪声的标准差。ALPHA 是调整参数，它必须是大于 1 的实数。其值越大，降噪信号或图像的小波包表示越稀疏。ALPHA 的典型值为 2。

2. wpthcoef 函数

【函数功能】小波包系数阈值处理。

【语法格式】T= wpthcoef (T,KEEPAPP,SORH,THR)

【使用说明】wpthcoef 是一维或二维小波包降噪和压缩的工具函数。它返回小波包树 T 的系数进行阈值处理后的新的小波包树 T。

如果KEEPAPP=1，低频系数没有进行阈值处理，否则进行了阈值处理。SORH='s'，表示软阈值，SORH='h'，表示硬阈值。THR是阈值。

3. wpdencmp函数

【函数功能】使用小波包进行降噪和压缩。

【语法格式】

① [XD,TREED,PERF0,PERFL2] = wpdencmp (X,SORH,N,'wname',CRIT,PAR,KEEPAPP)

② [XD,TREED,PERF0,PERFL2] = wpdencmp (TREE,SORH,CRIT,PAR,KEEPAPP)

【使用说明】该函数是一维或二维小波包压缩或降噪的导向函数，它使用小波包，对信号或图像执行降噪或压缩过程。小波包压缩和降噪的思想和过程与小波框架里面的压缩和降噪非常类似。

格式①对于输入信号 X（一维或二维），由小波包系数阈值处理后得到降噪或压缩后的XD。附加的输出变量TREED是XD的小波包最优树分解树。PERF0和PERFL2是能量恢复和压缩的L^2范数百分比。SORH ('s' or 'h')是软阈值或硬阈值。

小波包分解使用小波'wname'，在第N层执行，使用由字符串CRIT和参数PAR定义的熵函数进行最优分解。阈值参数也是PAR。如果KEEPAPP=1，低频系数不能进行阈值处理，否则可以。

格式②使用和上面相同的选项，具有同样的输出，但它是由被降噪或压缩信号输入小波包分解树TREE得到的。

另外，如果CRIT='nobest'，则没有进行优化。

9.1.11 小波工具箱其他应用函数简介

这里我们着重介绍4个函数，表9-11中其他一些树管理函数的使用比较容易，这里就不多做讲述。

1. wfbm函数

【函数功能】分数布朗运动合成。

【语法格式】

① FBM = wfbm (H,L)

② FBM = wfbm (H,L,'plot')

③ FBM = wfbm (H,L,NS,W),

④ FBM = WFBM (H,L,W,NS)

⑤ wfbm (H,L,'plot',NS,W)

⑥ WFBM (H,L,'plot',W,NS)

【使用说明】格式①根据Abry和Sellan提出的算法，返回由Hurst参数(0 < H < 1)和长度L决定的fractional Brownian motion信号FBM。

格式③和④使用NS重构步骤和正则正交小波W返回FBM。

格式wfbm (H,L,'plot',NS)或wfbm (H,L,'plot',W)或wfbm (H,L,'plot',NS,W)或wfbm (H,L,

'plot',W,NS)产生并画出 FBM 信号。

wfbm (H,L)等同于 WFBM (H,L,6,'db10')。

wfbm (H,L,NS)等同于 WFBM (H,L,NS,'db10')。

wfbm (H,L,W)等同于 WFBM (H,L,W,6)。

2. wfbmesti 函数

【函数功能】分数布朗运动参数估计。

【语法格式】HEST = wfbmesti (X)

【使用说明】该函数假设 H 来源于分数布朗运动的参数 H，返回 H 的 3 个估计 HEST 行向量。第 1 个估计基于二阶离散导数；第 2 个估计基于小波；第 3 个估计基于线性回归，并画出重对数图。

3. wfusmat 函数

【函数功能】融合两个矩阵或数组。

【语法格式】

① C = wfusmat (A,B,METHOD)

② [C,D] = wfusmat (A,B,METHOD)

【使用说明】格式由 METHOD 定义的融合方法返回矩阵 A 和 B 的融合矩阵 C。矩阵 A 和 B 必须有相同的规模，C 的维数和它们一样。可用的融合方法有简单方法，包括最大值、最小值、平均值、第一个元素、第二个元素或者随机元素融合 X1 和 X2 的低频和高频部分；还有依靠参数的方法，包括线性融合、上－下融合、下－上融合、左－右融合、右－左融合等。

4. wfusimg 函数

【函数功能】融合两幅图像。

【语法格式】

① XFUS = wfusimg(X1,X2,WNAME,LEVEL,AFUSMETH,DFUSMETH)

② [XFUS,TXFUS,TX1,TX2] =

wfusimg(X1,X2,WNAME,LEVEL,AFUSMETH,DFUSMETH)

wfbsimg(X1,X2,WNAME,LEVEL,AFUSMETH,DFUSMETH,FLAGPLOT)

【使用说明】基于小波分解的图像融合的思路是对两个原始图像的分解后的高频系数和低频系数进行融合。

格式①使用 WNAME 小波，对原始图像在 LEVEL 层 X1 和 X2 的分解进行融合，返回图像 XFUS。AFUSMETH 和 DFUSMETH 分别指定了低频系数和离频系数的融合方法。X1 和 X2 矩阵必须具有相同的规模。

格式②除了返回矩阵 US 以外，还返回类 WDECTREE 的 3 个对象，它们分别与 XFUS、X1 及 X2 相关联。wfusimg(X1,X2,WNAME,LEVEL,AFUSMETH,DFUSMETH,FLAGPLOT)还画出 XFUS、X1 及 X2 三个对象。

9.2 Matlab 小波分析的图形实现

在Matlab命令符下键入wavemenu后按回车键，即会出现小波工具箱主菜单窗口（Wavelet Toolbox Main Menu），如图9.1所示。

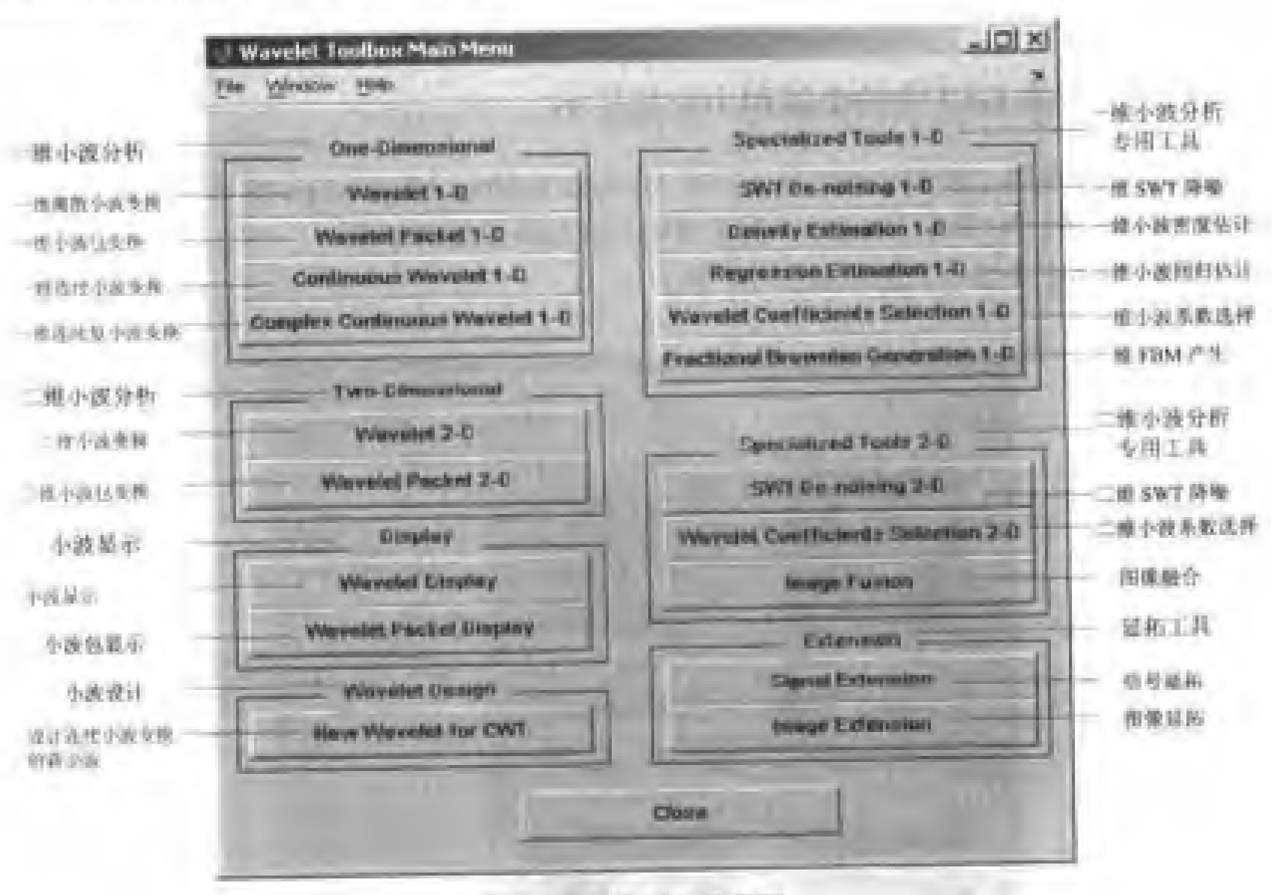


图9.1 小波GUI的界面

可以看出，小波GUI的功能非常丰富，我们进行了整理，如表9.15所示。

表9.15 小波GUI的主要功能

主要功能	子功能	主要功能	子功能
一维小波分析	一维离散小波变换	一维小波分析专用工具	一维平稳小波消噪
	一维小波包变换		一维小波变换密度估计
	一维连续小波变换		一维小波变换回归估计
	一维连续复小波变换		一维小波系数选取
二维小波分析	二维离散小波变换		一维FBM产生（新功能）
	二维离散小波包变换	二维小波分析专用工具	二维平稳小波消噪
小波显示	小波显示		二维小波系数选择
	小波包显示		图像融合

续表

主 要 功 能	子 功 能	主 要 功 能	子 功 能
小波构造	CWT 小波构造（新功能）	延拓工具	信号延拓
			图像延拓

这里，用于 CWT 的小波构造、一维 FBM 产生及图像融合是 Matlab 7 小波工具箱的新增功能。

9.2.1 小波 GUI 中的小波和小波包显示

小波 GUI 中的小波和小波包显示位于 Wavelet Toolbox Main Menu 窗口中的 Display 组件框中，单击它们可以显示 Matlab 中的小波和小波包函数簇的信息。

例如，dmer 小波信息显示如图 9.2 所示。

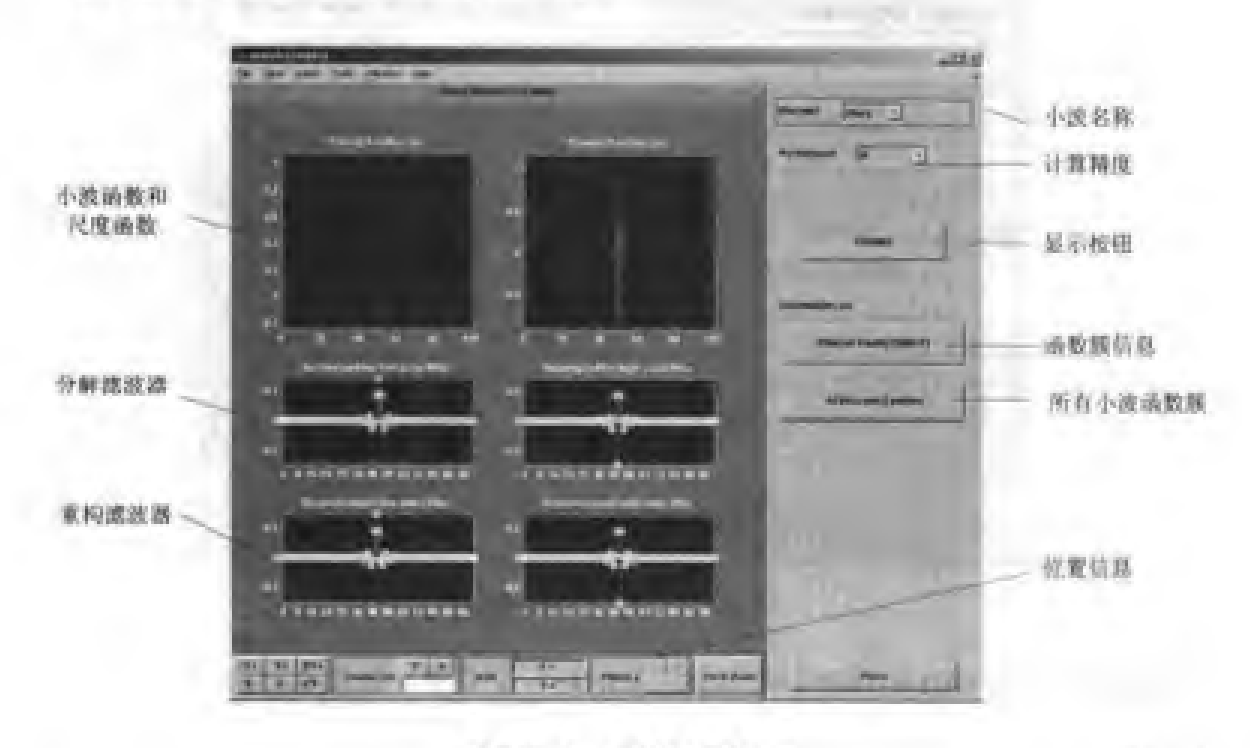


图 9.2 dmer 小波信息显示

图 9.2 中，Refinement 决定了小波计算的精度，这里函数是在 2^8 点上计算得到的。此值越大，计算精度越高。这里可显示的小波有：Haar wavelet、Daubechies wavelet、Biorthogonal wavelet、Reverse Biorthogonal wavelet、Coiflets wavelet、Symlets wavelet、Morlet wavelet、Discrete Meyer wavelet、Mexican Hat wavelet、Meyer wavelet 等实小波和 Complex Gaussian wavelet、Morlet wavelet、Frequency B-Spline wavelet、Shannon wavelet 等复小波。

单击【××Family】按钮（××表示所显示的小波名），可以显示该小波簇的详细信息。单击【All Wavelet Families】按钮，可以显示所有小波的信息。

小波包显示的主窗口如图 9.3 所示。

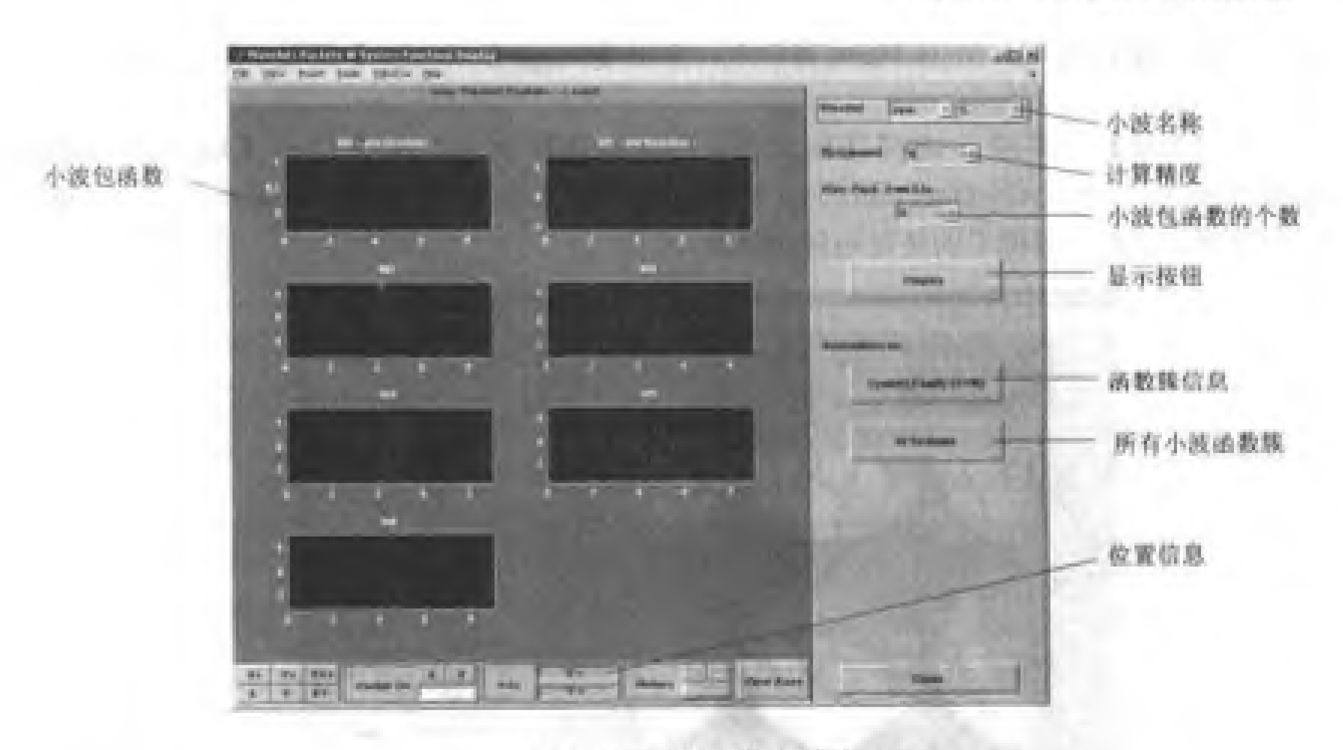


图9.3　sym小波包信息显示

9.2.2　小波GUI中的信号和图像延拓

单击【Signal Extension】按钮，可以进入信号延拓界面，如图9.4所示。



图9.4　信号延拓界面

装载数据：从【File】菜单中，选择【Load Signal】命令，选择演示的MAT文件 noisbloc.mat，它位于Matlab软件路径下的toolbox/wavelet/wavedemo。这是一个噪声信号，下面用“Signal Extension”工具对其进行延拓。

从 Wavelet Toolbox Main Menu 窗口中，单击【Image Extension】按钮，就进入了图像延拓工具界面。

从【File】菜单中，选择【Example Extension】命令，从中选取【facets-Extension Mode:ppd-Size[512 512]-Direction:[both both]】命令。延拓后的图像如图 9.5 所示。红线框中的是原图像，而黄线框中的是延拓后的图像。

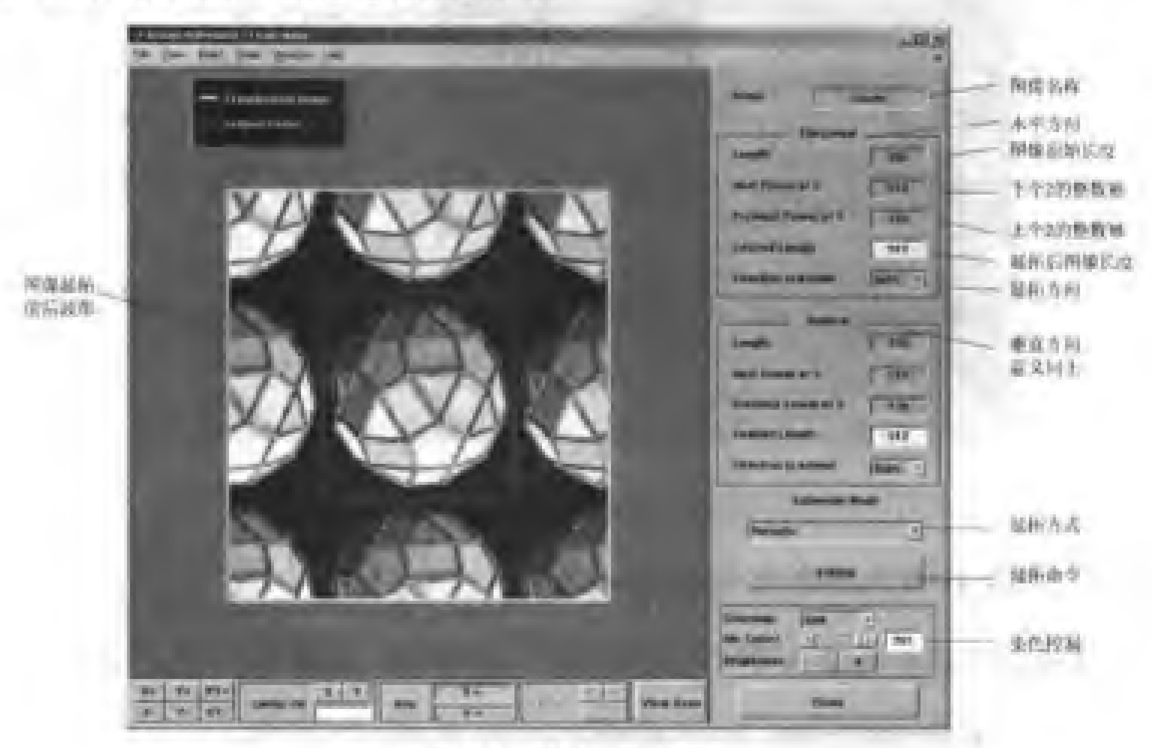


图 9.5　延拓前后的图像

在熟悉了信号延拓的基本原理和操作方法之后，读者对图像延拓应该不会陌生。这里主要有两个区别：一是在水平和垂直方向都要定义延拓后的长度，如果定义的长度小于原长度，那么即为图像裁剪；二是增加了染色控制模块。

退出该工具时，系统同样会询问是否保存延拓后的图像，读者可以根据自己的需要选择保存或放弃。

9.2.3　一维小波变换

该工具位于 Wavelet Toolbox Main Menu 窗口中的第一栏，它包括一维连续小波变换、一维连续复小波变换、一维离散小波变换及一维小波包变换。

1. 一维连续小波变换

通过 GUI，现在我们采用图形接口方式中的 "Continuous Wavelet 1-D" 工具来完成上面的分析过程。读者可以装载 Matlab 软件自带的信号，它们位于安装路径下的 toolbox/wavelet/wavedemo 下，也可以装载自己所要分析的工程信号。这里我们处于演示目的，使用的是示例中的信号，请选择【File】→【Example Analysis】→【with mexh at scales[1: 1:32]→frequency breakdown】命令。

选择好小波函数、尺度、采样周期、Coloration Mode（染色模式）等参数后，就可以单

击【Analyze】按钮执行一维连续小波变换。经过短时间的计算后将出现如图 9.6 所示的分析结果。

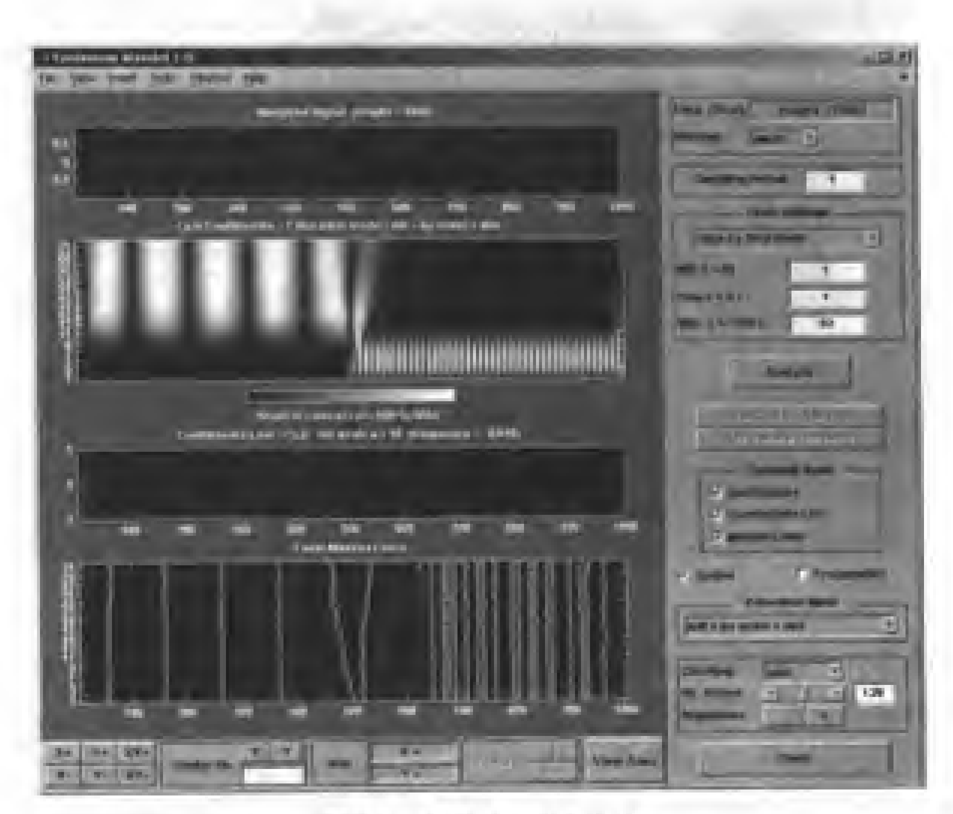


图 9.6　连续小波分析结果

通过图形用户界面，可以非常方便地观察小波系数线（Wavelet Coefficients Line）、观察最大值（Local Maxima Line）。选中 Frequencies 单选框，然后在系数图上单击鼠标右键，在底部的 Info 框中将指示出鼠标所在位置对应的位置信息（X）和频率信息（Frq，单位 Hz）。小波工具箱的这种在尺度和频率信息之间切换的功能可以让用户了解尺度和频率的对应关系，这种对应关系依赖于小波基函数和采样周期。另外，按住鼠标左键不放，在感兴趣的信号部分拖放一个矩形框，然后单击底部的【X+】按钮，即只在水平方向上放大。这样就会出现放大了的信号和对应的系数图。

采用一维连续复小波图形分析工具（Complex Continuous Wavelet 1-D）来完成一维连续复小波分析，选择【File】→【Example Analysis】→【with cgau2 at scales[1:1:64]→Test Singuralities(1)】菜单命令，就会出现复小波的分析结果。在小波工具箱提供的图形用户界面 GUI 中，共有 4 种复小波，即 cgau、shan、fbsp 和 cmor。

2. 一维离散小波变换

在小波工具箱 GUI 主菜单中单击【Wavelet 1-D】按钮，就会出现一维离散小波分析图形工具。使用 Matlab 7 软件自带的 noissin 信号（位于安装路径下的 toolbox/wavelet/wavedemo 下），为了执行多尺度一维分解，选择右上角的 Level 值为 3，Wavelet 为 haar，单击【Analyze】按钮来完成前面在命令行方式中实现的三尺度分解，得到分解的低频信号 a_3，高频信号 d_3，d_2，d_1，结果如图 9.7 所示。

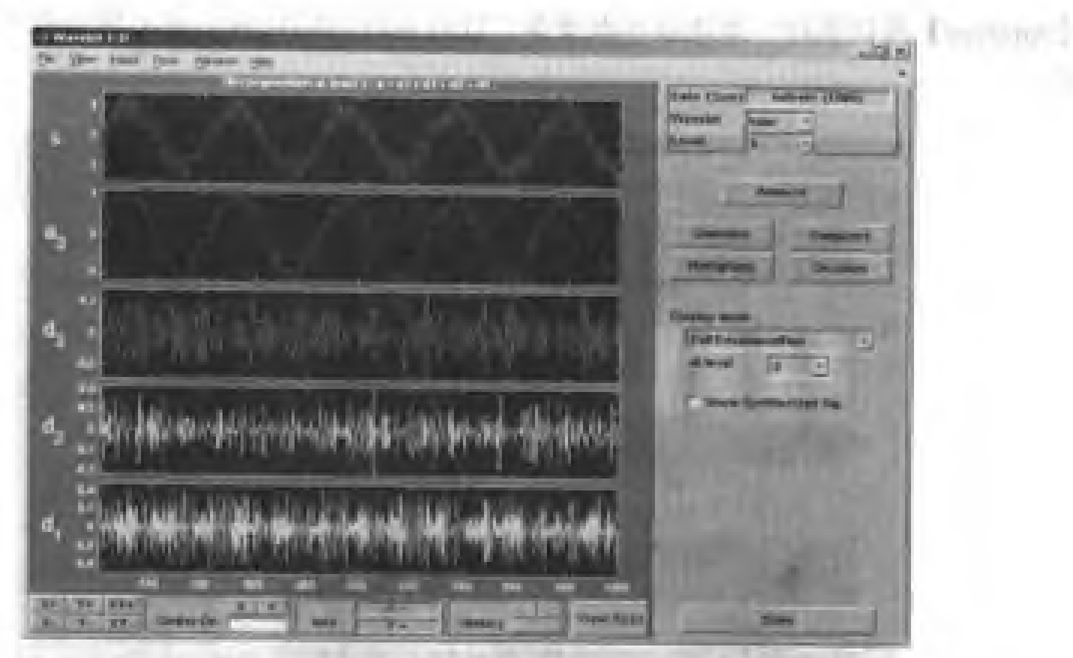


图 9.7 三尺度分解

在默认情况下，一个信号用小波分解后，显示模式是完全分解模式（Full Decomposition）。除此之外，Matlab 7 小波工具箱还提供了另外几种显示模式，用户可以在 Display Mode 列表框（显示模式）中进行选择。共有以下 6 种模式：分别是 Full Decomposition（完全分解模式）、Show and Scroll（显示滚动模式）、Stem Cfs（柱状系数图模式）、Separate Mode（分离模式）、Superimpose Mode（叠加模式）和 Tree Mode（树模式）。

基于离散小波变换的结果，我们可以实现信号的统计、压缩和去噪等功能。

图形工具的预定义阈值去噪方式可以很容易地将噪声从信号中去除。单击信号小波分解后窗口右边出现的【De-noise】按钮，出现相应的 Wavelet 1-D -- De-noising 窗口，如图 9.8 所示。

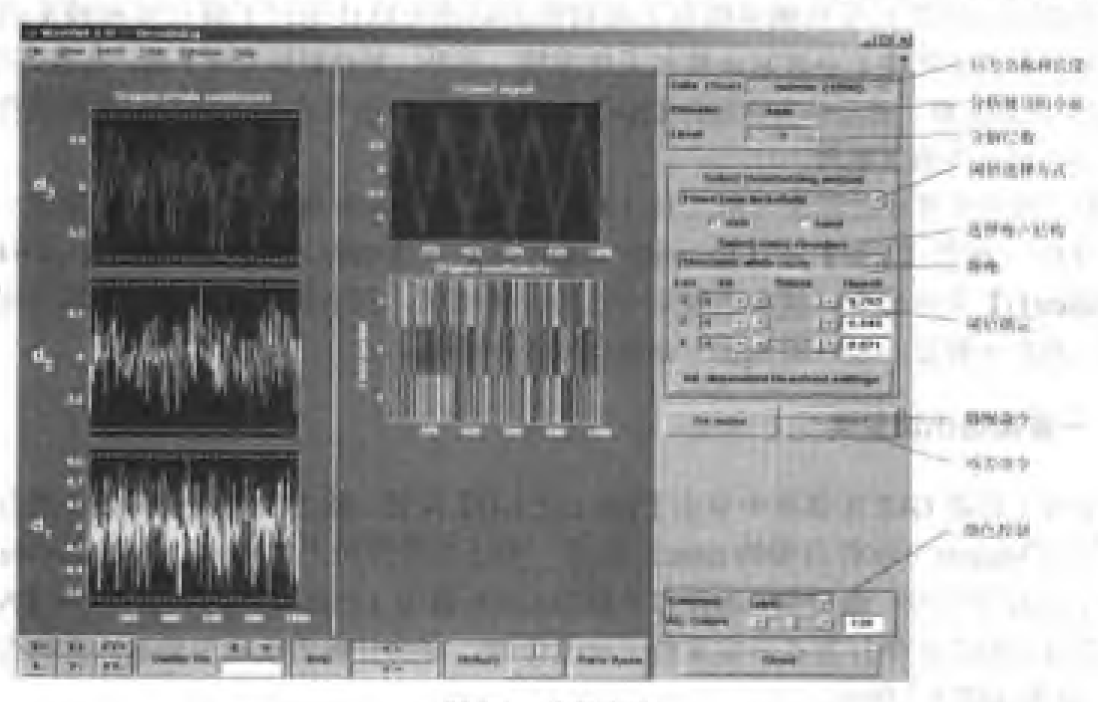


图 9.8 信号去噪

比如，选择 soft thresholding（软阈值）和 unscaled white noise（尺度未知的白噪声）后，单击【De-noise】按钮就可以实现信号的去噪，同时分析消噪后的信号和原信号相比的残差，在 Wavelet 1-D -- De -noising 窗口中单击【Residuals】按钮，出现 More on Residuals for Wavelet 1-D -- De -noising 窗口，可以看出，除均值、模式、中值、误差、标准差等统计量外，时间序列图（自相关和自谱）及频域分布图也显示出来。

对于信号压缩，小波图形工具箱GUI提供了自动阈值(Global thresholding)和手动阈值(By level thresholding）两种方式进行压缩。单击【Compress】按钮，就会出现数据压缩窗口。比如，选择统一阈值，此时阈值是一个全局阈值，可在 Select Global Threshold 选择框中拖动滚动条或在文本框中输入数值来选择该阈值，也可在左边的 Automatic thresholding 显示方框中用鼠标拖动绿色的阈值虚线来选择。压缩后的结果如图 9.9 所示。

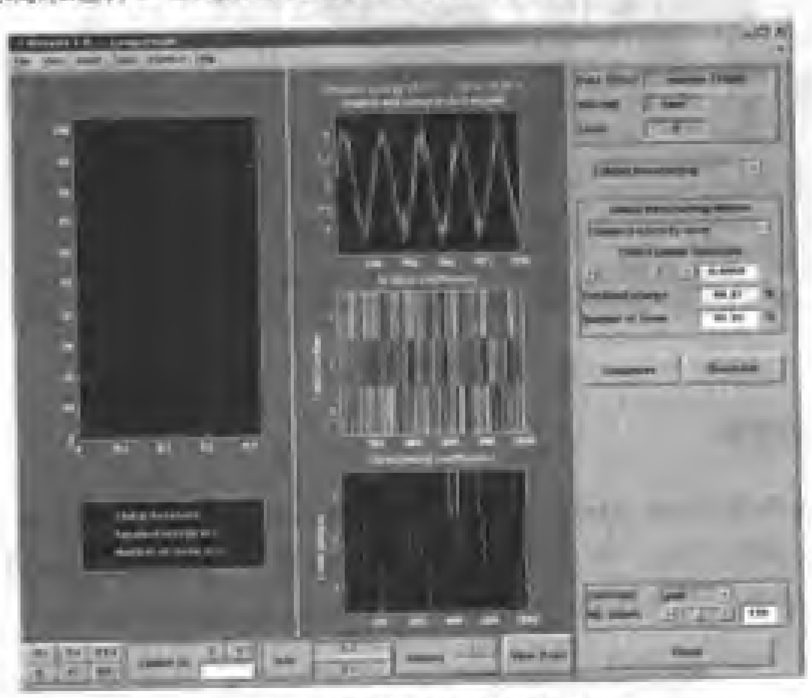


图 9.9 压缩信号和原始信号

可以看出，压缩后大部分噪声被去除了，但保留了原信号的 89.37%的能量，置 0 系数的百分比为 85.20%。接下来同样可以分析压缩后的信号和原信号相比的残差。

另外，当进行按层确定阈值时，用户可以对高频系数的每一层都选择不同的阈值，每层阈值的选择方法和上述类似。

对于信号统计，在 Wavelet 1-D 窗口中单击【Statistics】按钮，就会出现一维离散小波统计窗口。一维离散小波变换 GUI 提供了 4 种信号统计功能，包括原信号统计、合成信号统计、近似信号统计及细节信号统计，分别对应于窗口右边的 4 个不同的单选框：Orignal signal、Synthesized signal、Approximation 和 Detail，选择其中某个按钮，则在左边的显示区域中显示相应信号的统计结果，包括文本方式显示的计算值、以图形方式显示的直方图和以求和形式显示的直方图。当选中 Approximation 和 Details 单选框后，在窗口中部会出现 Approximations Levels 和 Details Levels 的选择框，让用户具体选择要显示的低频系数或高频系数的层次。Number of bins（柱条数）文本框可以让用户输入两种直方图中显示的柱条数量。这里我们选择 Approximation 单选框和 Level 1，然后单击【Show statistics】按钮，统计结果

如图 9.10 所示。

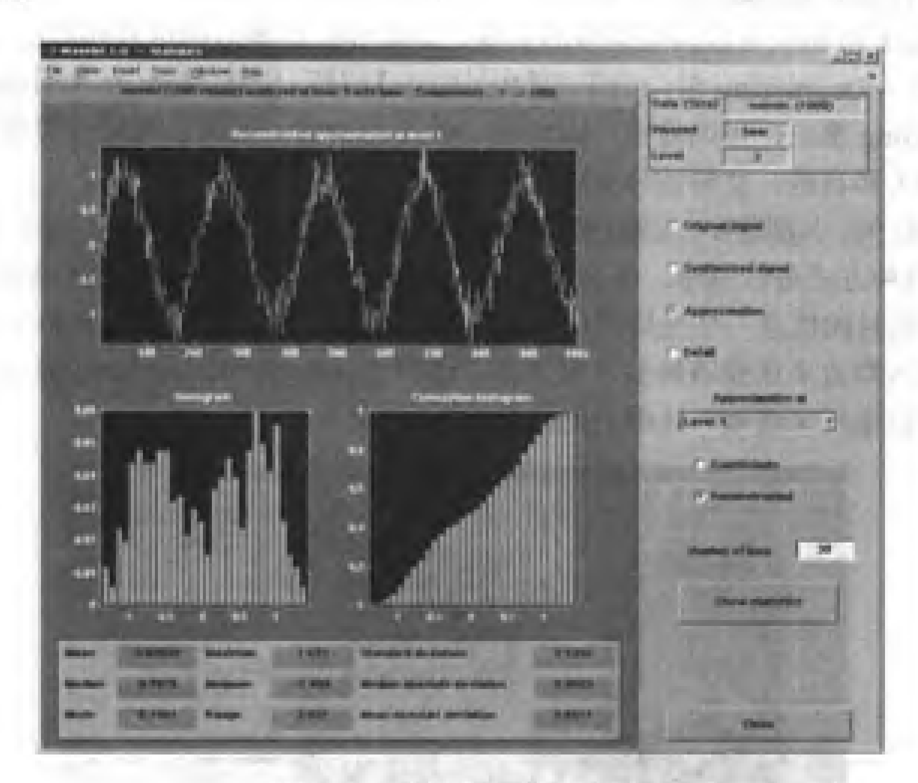


图 9.10　低频信号统计结果

3. 一维小波包变换

在 Wavelet Toolbox Main Menu 窗口中单击【Wavelet Packet 1-D】按钮，出现如图 9.11 所示的一维小波包图形工具。在主菜单中选择【File】→【Load Signal】菜单命令，选择 noissin.mat 文件，结果如图 9.11 所示。

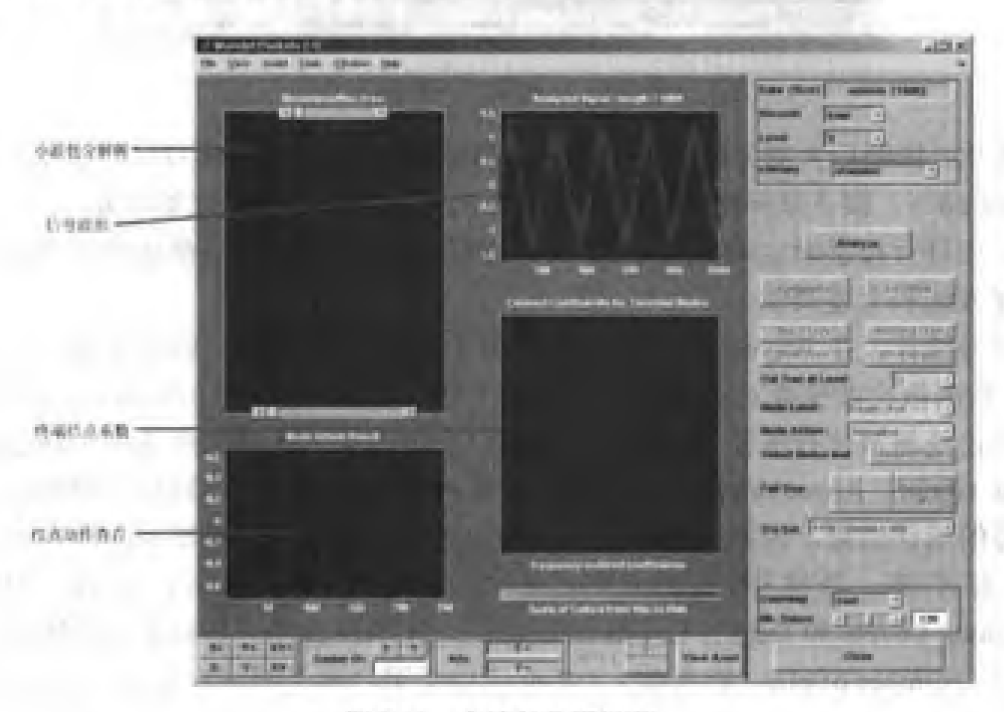


图 9.11　小波包分析界面

一维小波包图形工具窗口与前面我们用过的一维小波图形工具类似，左边是信号的显示区域，右边是对信号进行小波包分析的各种按钮和参数选择设置框。在左边的显示区域中有4个图形区：

（1）"Analyzed Signal" 在右上角，显示原始信号波形。

（2）"Decomposition Tree" 在左上角，显示小波包分解树结构。右移上端的滚动条可以放大图形，左移滚动条则缩小图形；右移下端的滚动条可以将图形往右移，左移滚动条则可以将图形往左移。当对信号进行多层小波包分解后的树结构很复杂时，用户可以通过这两个滚动条方便地查看各个结点的细节。

（3）"Node Action Result" 在左下角，显示某个结点分析的结果。

（4）"Colored Coefficients for Terminal Nodes" 在右下角，显示信号小波包分解后系数的灰度图像。在一维小波包图形工具中，对一个信号是按照完整二叉树的方式来进行 n 层的小波包分解，这样终结点有 2^n 个。图形工具对第 n 层的 2^n 组分解系数以灰度图像的方式来显示其结果。

在图9.11窗口的右边可以对信号分析进行合理的设置。小波包基函数 Wavelet 下拉列表框选取 haar，分解层次 Level 取 3，Entropy（熵标准）下拉列表框选取 shannon，然后单击【Analyze】按钮，该按钮执行的功能是对信号进行小波包分解并将分解结果显示在窗口中，如图9.12所示。

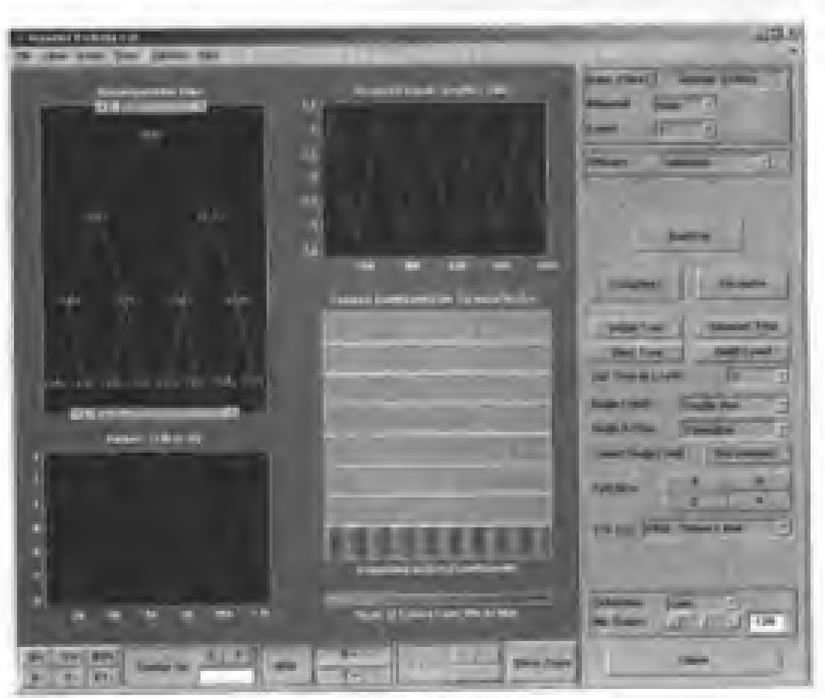


图9.12 小波包分解结果

单击 Full Size 中的【1】、【2】、【3】和【4】按钮，则放大显示它们对应于左边的图形，再单击一次，又重新回到原界面。

由于从小波包分解树中重构原始信号有许多可选的方法，所以在压缩信号之前我们先根据上面的熵标准计算出初始树的最佳子树。在图9.12中右上方单击【Best Tree】按钮，其功能就是在执行信号的小波包分解之后用 Entropy 下拉列表框确定的熵标准计算出最佳的小波包分解树。

【Initial Tree】按钮的作用是进行一个信号的小波包分解的初始树并显示。读者可以利用这两个按钮对初始树和最佳树进行对比。同样，单击【Best Level】按钮可以计算小波包分解的最佳层。单击【Wavelet Tree】按钮可以查看小波树。

同样可以利用小波包进行信号压缩。在信号小波包分解后对各层的高频系数进行阈值量化处理，然后对根据处理后的系数进行信号重构，从而达到对信号用小波包方法进行压缩的目的。进行阈值量化时，熵标准由 Entropy 下拉列表框确定。在一维小波包图形工具中，只有全局阈值。单击【Compress】按钮，会出现一维小波包压缩窗口。

读者可以根据实际需要，确定键入阈值的大小，或者直接在图中代表阈值的黄点线上按住鼠标左键不放，拖动该线来更直观地调整阈值以观察置零系数百分比和能量保留百分比的变化情况，如果一次不能达到理想的效果，可以多次重复。单击【Residuals】按钮，可以查看压缩后信号和原信号的残差。

利用小波包可以进行信号消噪。比如，采用一维小波包图形工具来对上面的 noissin 信号进行消噪。这里小波包基函数下拉列表框选取 haar，分解层次 Level 取 3，Entropy（熵标准）下拉列表框选取 log energy。针对图 9.12 的分析结果，进行小波包信号消噪。

首先单击【Best Tree】按钮，计算出最佳树以使下面的消噪处理更有效。然后再单击【De-noise】按钮，消噪结果如图 9.13 所示。

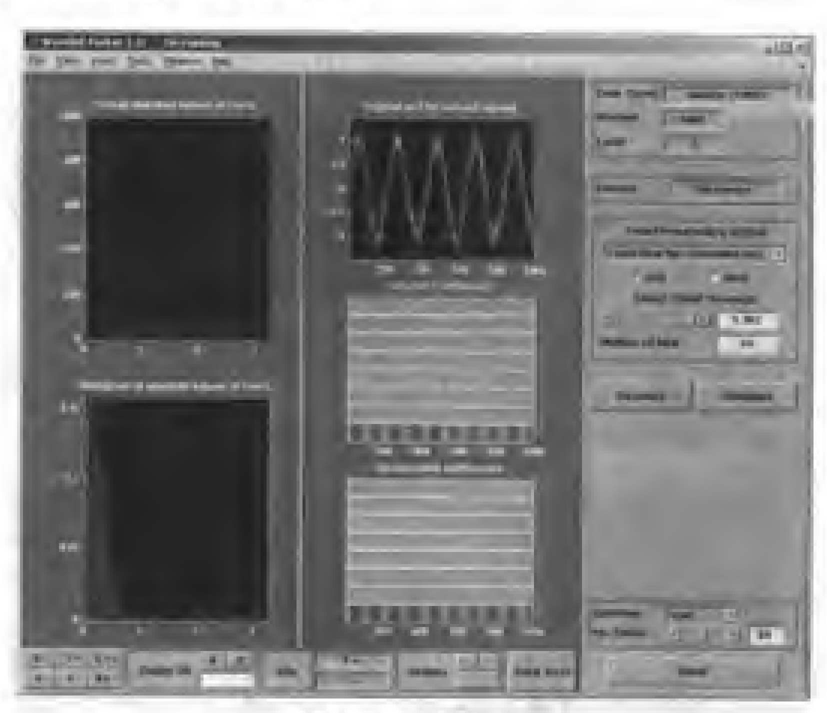


图 9.13 消噪结果

图 9.13 中红色的是原始信号，黄色的是消噪后的信号。单击【Residuals】按钮，可以查看消噪后的残差。

4. 一维小波变换专用工具

一维小波变换专用工具包括一维平稳小波降噪、一维小波密度估计、一维回归估计、一维小波系数选取和一维 FBM 信号产生。

（1）一维平稳小波降噪

在 Wavelet Toolbox Main Menu 窗口中单击【SWT De-noising 1-D】按钮，即出现一维离散平稳小波分析图形工具。执行【File】→【Load】→【Signal】菜单命令，选择 Matlab 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 noisbloc.mat 文件。注意 SWT 的信号长度必须是 2 的幂次的整数倍，如果不满足此条件，必须进行信号延拓。降噪之前需要对信号进行一维小波分解。选择基本小波为 haar，尺度 Level 为 5，单击【Decompose Signal】按钮，经过短时间的计算后将出现如图 9.14 所示的分解结果。

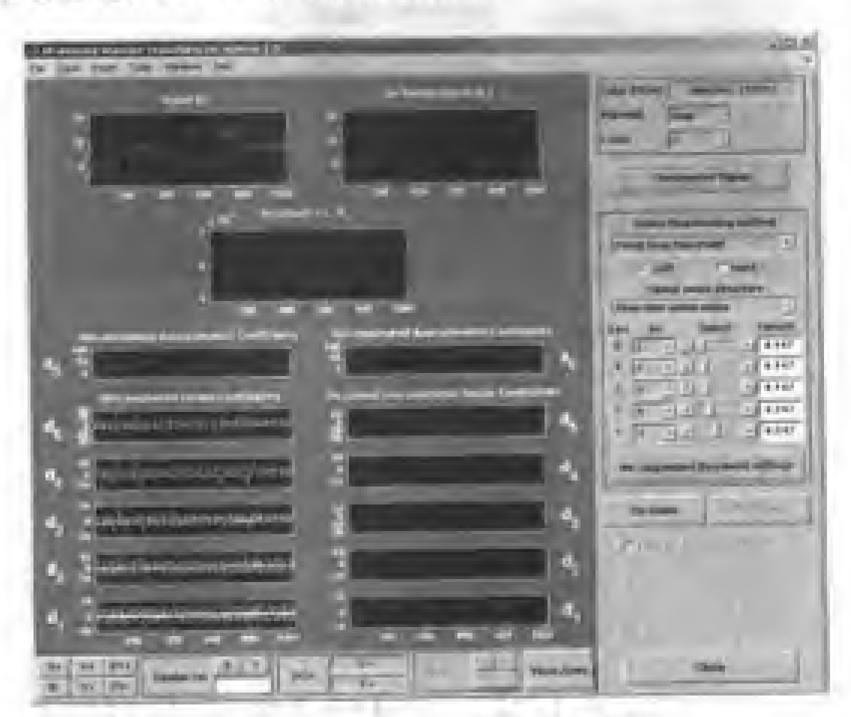


图 9.14 对信号进行一维平稳小波变换

选择 Fixed form soft threshold（固定软阈值）和 unscaled white noise（尺度未知的白噪声），并确定好阈值后，单击【De-noise】按钮，可以利用平稳小波变换来消噪。

（2）一维小波密度估计

在 Wavelet Toolbox Main Menu 窗口中单击【Density Estimation 1-D】按钮，出现一维小波密度估计图形工具。执行菜单命令【File】→【Load Data for Density Estimate】，选择 noissin.mat 文件。在图中的右上角选择基本小波为 sym6、分解尺度数 Level 为 4。选择好以上参数后，就可以单击【Decompose】按钮，经过短时间的计算后将出现分解结果。

采用默认的全局软阈值，结果以黄点线反映在窗口左边的图形中。选择好阈值后，单击【Estimate】按钮，结果如图 9.15 所示。当然也可根据实际情况通过窗口右边的滑动条来指示和调整相应各层次的阈值。

（3）一维回归估计

在 Wavelet Toolbox Main Menu 窗口中单击【Regression Estimation 1-D】按钮，出现一维回归估计图形工具。它包括一维等间距观测估计（确定性设计 Fixed Design）和一维随机间距观测估计（随机性设计 Stochastic Design）两种估计方式。

回归估计的基本步骤如图 9.16 所示。



图 9.15　密度估计 1

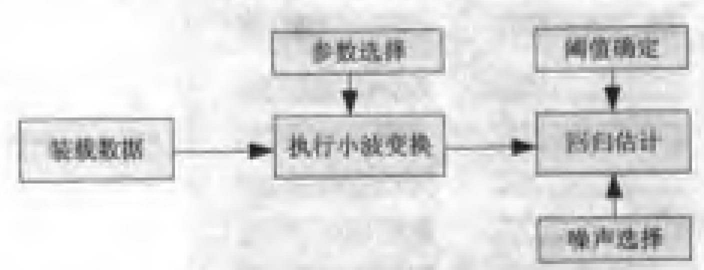


图 9.16　回归估计步骤

比如，在【Regression Estimation 1-D】工具的【File】菜单里执行【Load Data for Stochastic Design Regression】命令，仍以 noissin 信号作为原数据，结果如图 9.17 所示。

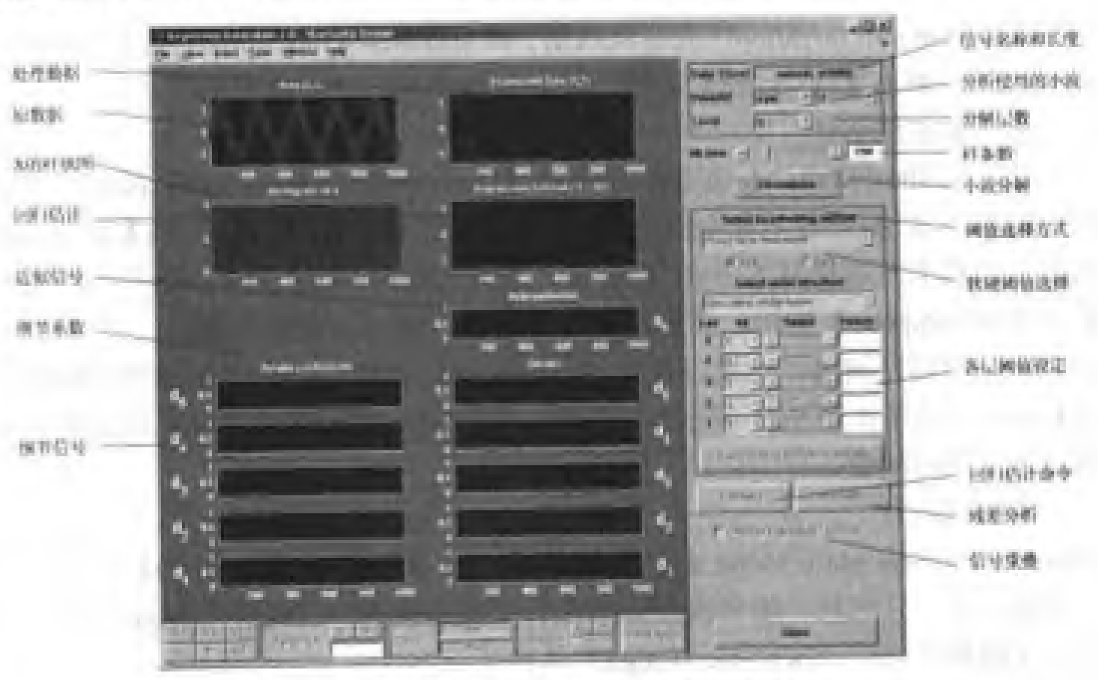


图 9.17　一维随机间距观测回归估计图形工具

（4）一维小波系数选取

在 Wavelet Toolbox Main Menu 窗口中单击【Wavelet Coefficient Selection 1-D】按钮，则出现一维离散小波系数选择窗口。执行【File】→【Load】→【Signal】菜单命令，选择 noissin.mat 文件。在图中的右上角进行选择，使基本小波为 haar，尺度数 Level 为 5。选择好以上参数后，单击【Analyze】按钮，经过短时间的计算后将出现 5 层的分解结果。在原始信号下面是它的小波分解低频系数 A5 和高频系数 D5、D4、D3、D2、D1。在合成信号（synthesized signal）下面是选取的小波系数，在这一步和左边的初始小波系数相同，因为还没有对初始小波系数进行选取，保留了所有的初始小波系数。

在 Define Selection method 下拉列表框里，选取小波系数共有 4 种方式：Global（全局选取）、By Level（按层选取）、Manual（手动选取）和 Stepwise Movie（分步动画）。系统在装载信号时，系统会自动产生各系数的个数，并显示于各编辑框中。窗口右边的 App.cfs（Approximation Coefficients）下拉列表框里为低频系数的 3 种选择方式，Select All 表示全部选择，Unselect 表示全部舍弃，还可以选择 Selectable，在最大值系数选取个数编辑框里输入新值并回车，然后单击【Apply】按钮，就可以保留部分低频小波系数。

了解了以上的系数选取方法之后，我们就可以很方便地得到选择后的系数，这里我们采用“Global”和“Select All”，并将总数变为 251，得到的结果如图 9.18 所示。读者可以通过单击【Residuals】按钮分析残差，这里由于篇幅限制，就不再介绍了。

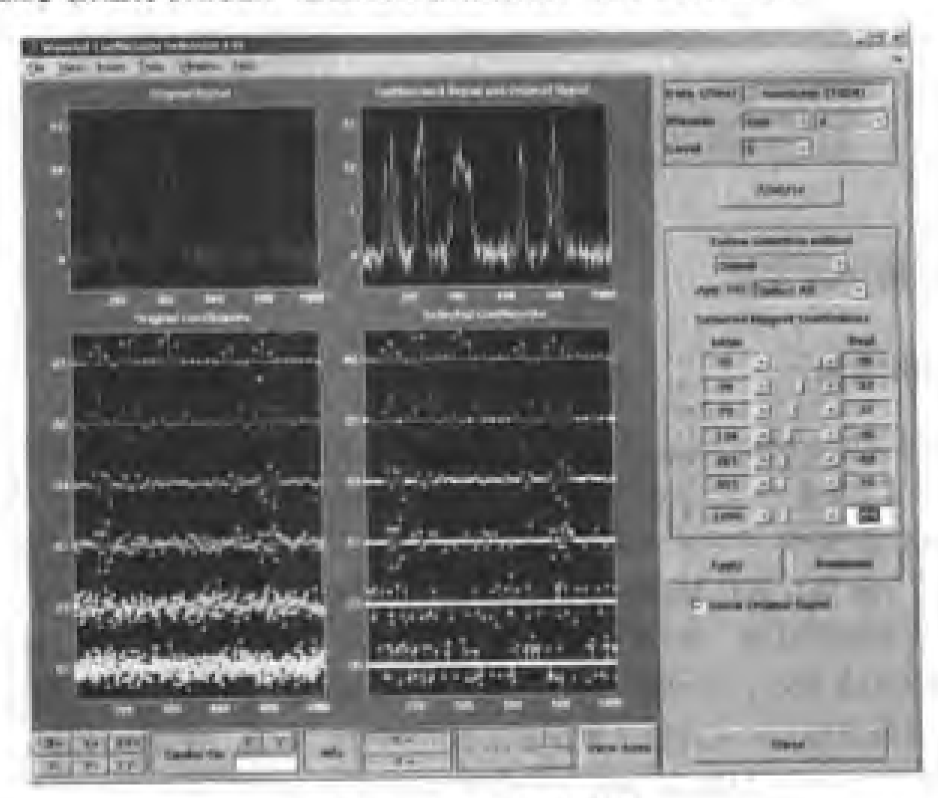


图 9.18 小波系数选取结果

（5）一维 FBM 信号产生

FBM（Fractional Brownian Motion）是连续时间高斯过程，它决定于参数 H（hurst）：$0<H<1$。$H=0.5$ 时，它产生 Brownian 运动，它的导数是白噪声。FBM 的分布是自相似的，其增量方差由式

$$Var(fBm(t) - fBm(s)) = v\,|t-s|2H$$

给出。这里 v 是正的常数。

FBM 信号可以通过 Matlab 小波工具箱的命令行和 GUI 产生，这里我们介绍 GUI 方法。

在 Wavelet Toolbox Main Menu 窗口中单击【Fractional Brownian Generation 1-D】按钮。出现一维 FBM 合成工具，这里，我们使用 db10 小波，精度选 10，信号长度选 1000，Hurst 参数选 0.6，单击【Generate】按钮即产生了 FBM 信号，如图 9.19 所示。



图 9.19　产生的 FBM 信号

9.2.4　二维小波变换

1. 二维小波分解

在 Wavelet Toolbox Main Menu 窗口中单击【Wavelet 2-D】按钮，出现二维离散小波分析图形工具。执行【File】→【Load】→【Image】菜单命令，选择 Matlab 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 belmont2.mat 文件。这里支持的图像格式有 mat、bmp、hdf、jpg、jpeg、pcx 等，这是 Matlab 小波工具箱 3.0 的最新支持。

在图中右上角进行选择，使分解的基本小波为 haar，尺度数 Level 为 3。选择好以上参数后，单击【Analyze】按钮，经过短时间的计算后将出现如图 9.20 所示的分解结果。

二维小波分解图像工具可进行 4 种分析，包括统计、压缩、直方图和降噪。它们分别由【Statistics】、【Compress】、【Histograms】和【De-noise】4 个按钮来执行。

2. 二维小波包分解

在 Wavelet Toolbox Main Menu 窗口中单击【Wavelet Packet 2-D】按钮，出现二维小波包图形工具。这里，我们通过执行【File】→【Load】→【Image】菜单命令，选择 belmont2.mat 文件，进行小波包分析。二维小波包分解与一维小波包图形工具很相似，只不过处理的是

图像。

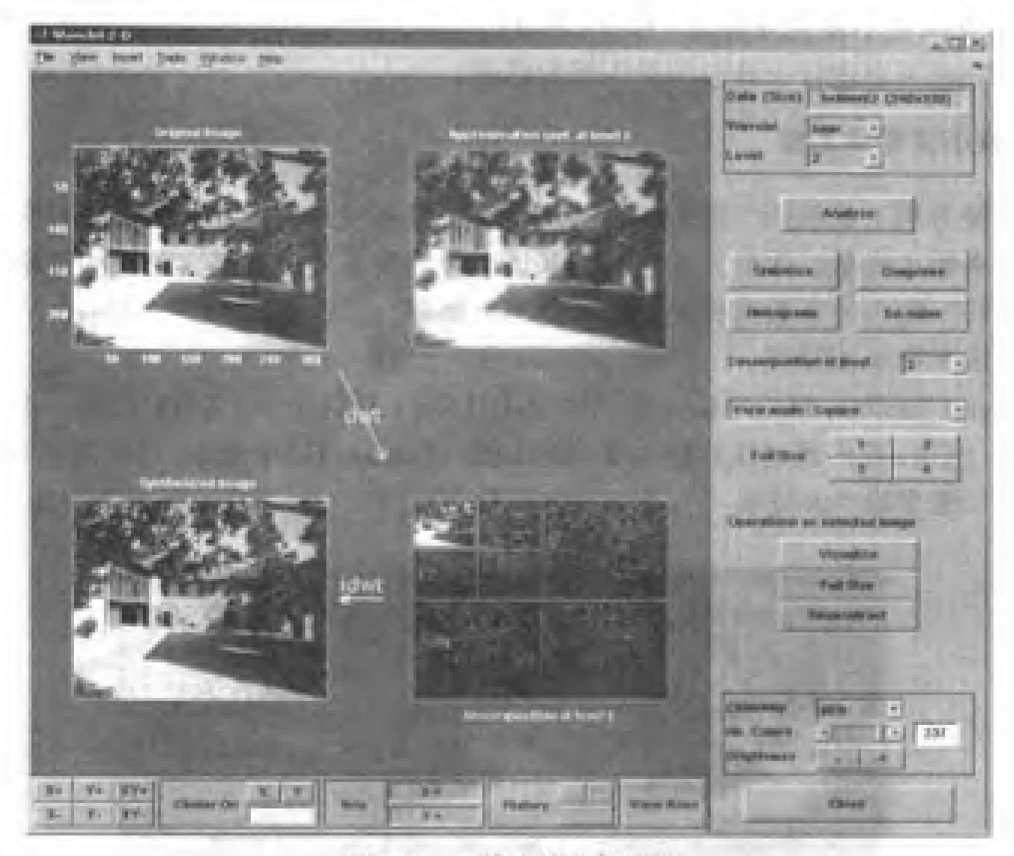


图 9.20　二维小波分解结果

二维小波包分解不仅对低频部分进行分解，而且对水平、垂直和对角 3 个方向也进行分解，所以每个父结点都对应 4 个子结点，分解后的结果如图 9.21 所示。图中对各个空间和显示区都进行了说明，这里就不多做介绍了。

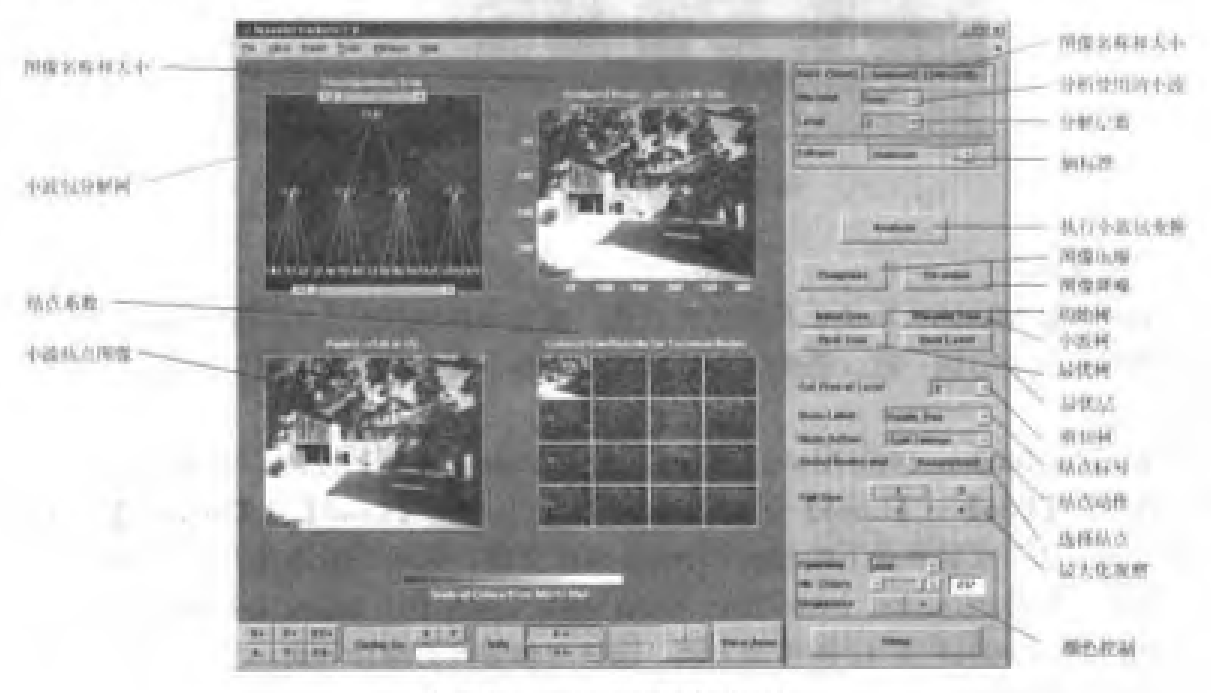


图 9.21　二维小波包图形工具

采用二维小波包工具，可以计算初始树（Initial Tree）、小波树（Wavelet Tree）、最佳树（Best Tree）和最佳小波包分解层（Best Level）。

另外，同样可以进行小波包图像压缩和降噪。

3. 二维小波专用工具

二维小波专用工具包括二维平稳小波降噪、二维小波系数选择和图像融合 3 大功能。其中二维小波系数选择是以图形方式来探究基于多种小波系数选取方案的重构策略，由于该图形工具基本和一维小波系数选择类似，所以不再解释。

（1）二维平稳小波降噪

在 Wavelet Toolbox Main Menu 窗口中单击【SWT De-noising 2-D】按钮，出现二维小波包图形工具。这里，我们通过执行【File】→【Load】→【Image】菜单命令，同样选择 belmont2.mat 文件，进行二维平稳小波降噪。

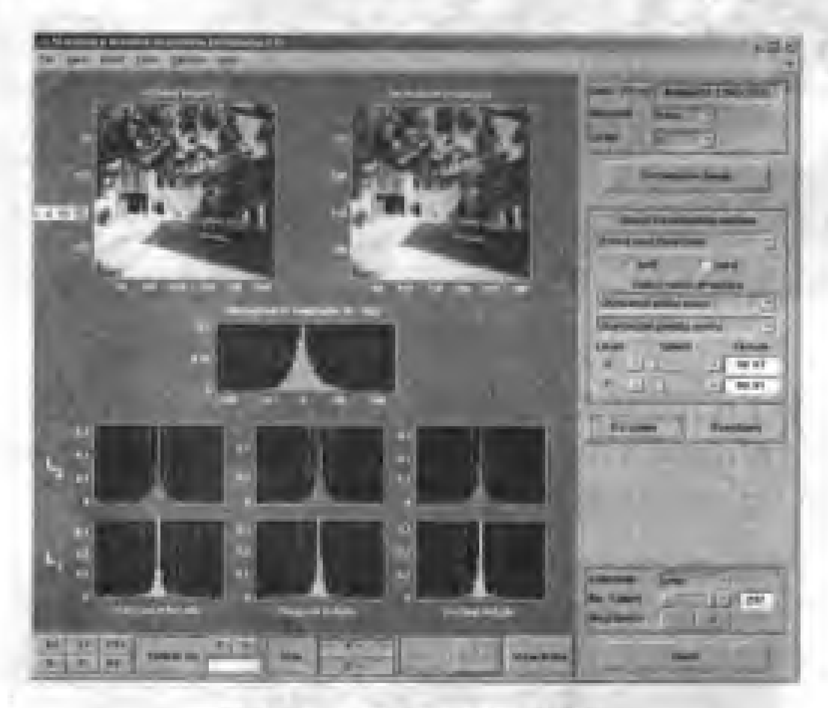


图 9.22　二维离散平稳小波降噪

（2）图像融合

图像融合是 Matlab 7 小波工具箱的新功能，就是将两幅图像[illegible]新的图像，当然这两幅图像类型和大小必须一致。图像融合可以通过命令行和 GUI 来实现，这里我们介绍 GUI 的实现方法。

在 Wavelet Toolbox Main Menu 窗口中单击【Image Fusion】按钮，出现图像融合工具。然后先后执行【File】→【Load】→【Image1】和【File】→【Load】→【Image2】菜单命令，同样选择 Matlab 7 安装目录下的 bust 和 musk 两个文件，这两个图像大小都是 256×256。

图像融合共有 max、min、mean、rand、linear、UD_fusion、DU_fusion、LR_fusion、RL_fusion、img1、img2 和 userDEF 等 12 种可选方式。选择两幅图像的融合方式，单击【Apply】按钮即可以对图像进行融合。

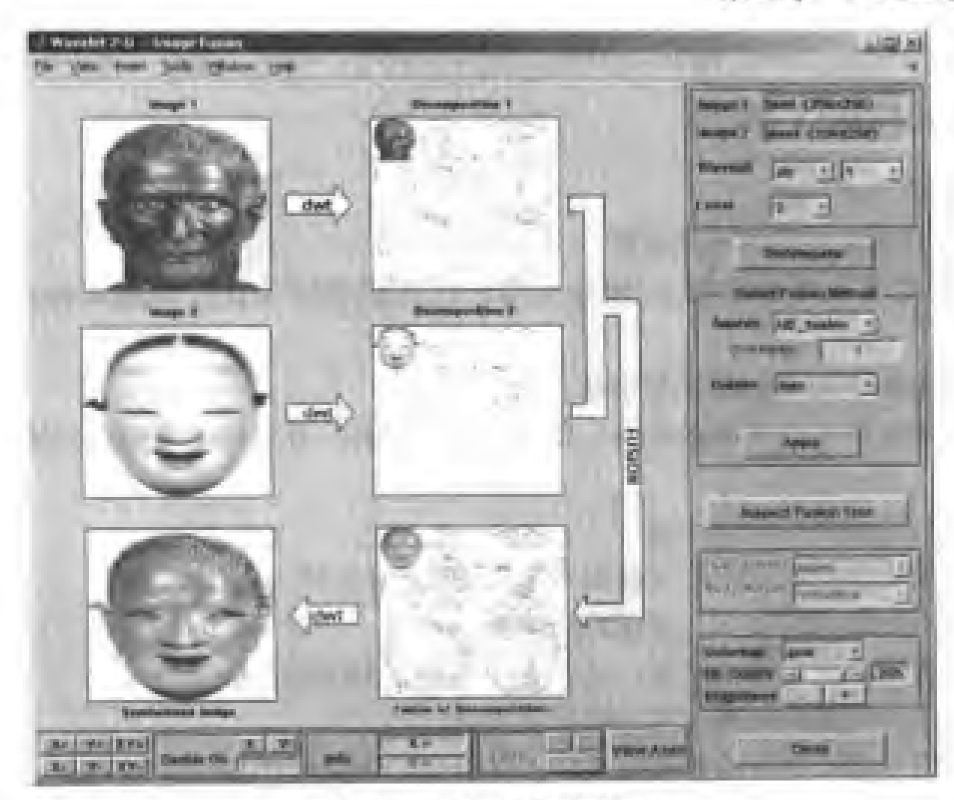


图 9.23　图像小波融合结果

单击【Inspect Fusion Tree】按钮可以查看融合树。在融合树中，可以方便地观察每个树结点的图像。

9.3　自定义小波基函数的添加

在实际应用中，有时这些小波并不能满足用户特定的分析需求，必须使用用户自己定义的小波函数，这就面临一个如何将用户自己定义的小波函数添加到小波工具箱里的问题。用户可以利用系统通过的小波管理函数 wavemngr 来完成这一功能，系统允许用户采用命令窗口方式和图形工具方式这两种方式来定义新的小波函数。

9.3.1　准备工作

wavemngr 函数可以允许添加新的小波函数和小波类，但是在添加之前，必须完成以下准备工作：

1. 选择小波类的全名（FN）

小波类的全名必须是一个字符串，已经定义的小波类名包括 Harr、Daubechies、Symlets、Coiflets、BiorSplines、ReverseBior、Meyer、Dmeyer、Gaussian、Mexican_hat、Morlet、Complex Gaussian、Shannon、Frequency B-Spline 和 Complex Morlet。

2. 选择小波类的缩写名（FSN）

小波类的缩写名必须是不大于 4 个字母的字符串，已经定义的小波类缩写名包括 harr、

db、sym、coif、bior、rbio、meyr、dmey、gaus、mexh、morl、cgau、fbsp 和 cmor。

3. 确定小波类型（WT）

共有 5 种类型的小波：

（1）利用 FIR 滤波器的正交小波

这类小波通过尺度滤波器 w 来定义的，已有的包括 Harr、Daubechies、Coiflets 和 Symlets。

（2）利用 FIR 滤波器的双正交小波

这类小波通过两个尺度滤波器 wr 和 wd 来定义的，分别用于重构和分解，已有的包括 BiorSplines。

（3）利用尺度函数的正交小波

这类小波通过小波函数和尺度函数来定义的，已有的如 Meyer。

（4）不利用 FIR 滤波器和尺度函数的小波

这类小波通过小波函数来定义，已有的如 Morlet 和 Mexican_hat。

（5）不利用 FIR 滤波器和尺度函数的复小波

这类小波也是通过小波函数来定义，已有的如 Complex Gaussian 和 Shannon。

4. 定义小波在给定类中的序号（NUMS）

如果一类中含有多个小波，那么缩写名加上其序号就成为每个小波函数的名字，参数 nums 是一个表示序号的字符串，比如对 Daubechies 小波：

FSN = 'db'

Nums = '1 2 3'

从而产生 3 种小波函数：db1、db2 和 db3。

5. 建立一个 MAT 或 M 文件（FILE）

函数 wavemngr 需要一个“file”参数，它是一个包含 MAT 或 M 文件名的字符串。如果一类中含有多个小波，则必须定义一个 m 文件，而且它的具体形式取决于小波的类型；如果一类中只有一个小波，则可以定义一个 mat 文件。具体地，M 文件的格式如下：

（1）第一类（利用 FIR 滤波器的正交小波）

M 文件的第一行必须是：

function w=file (wname)

其中输入参数 wname 是一个包含小波名的字符串，输出参数 w 是相应的尺度滤波器，且它必须是奇数长度，否则会补零。已有的比如 dbwavf.m 对应 Daubechies，coifwavf.m 对应 Coiflets。

（2）第二类（利用 FIR 滤波器的双正交小波）

M 文件的第一行必须是：

fuuction [wr,wd]=file (wname)

其中输入参数 wname 是一个包含小波名的字符串，输出参数 wr 和 wd 分别是相应的重构和分解尺度滤波器，且它们也必须是奇数长度的。已有的比如 biorwavf.m 对应 BiorSplines。

（3）第三类（利用尺度函数的正交小波）

M 文件的第一行必须是：

function [phi,psi,t]=file (lb,ub,n,wname)

返回尺度函数phi和小波函数psi在t时的值，t在[lb ub]上n点均匀取值。已有的如meyer.m对应 Meyer。

（4）第四类或第五类（不利用 FIR 滤波器和尺度函数的小波）

M 文件的第一行必须是：

function [psi,t]=file (lb,ub,n,wname)

返回小波函数 psi 在 t 时的值，t 在[lb ub]上 n 点均匀取值。已有的如 shanwavf.m 对应 Shannon，cmorwavf.m 对应 Complex Morlet。

6. 对不利用 FIR 滤波器的小波：定义有效支撑域

对第 3～5 类小波，需要定义有效的支撑域，也即定义上下限 lb 和 ub。

9.3.2 添加一个新小波函数

为了添加一个新的小波，wavemngr 命令有两种形式：

wavemngr('add',FN,FSN,WT,NUMS,FILE)

wavemngr('add',FN,FSN,WT,NUMS,FILE,B)

为了说明怎样使用 wavemngr 命令，表 9.16 给出了已有的小波类定义实例。

表 9.16　　小波类定义举例

类 型	语 法
1	wavemngr('add', 'Ndaubechies', 'ndb',1, '1 2 3 4 5', 'dbwavf')
2	wavemngr('add', 'Ndaubechies', 'ndb',1, '1 2 3 4 5 **', 'dbwavf')
3	wavemngr('add', 'Nbiorwavf', 'nbio',2, '1.1 1.3', 'biorwavf')
4	wavemngr('add', 'Nmeyer', 'nmey',3, '', 'meyer',[-8,8])
5	avemngr('add', 'Nmorlet', 'nmor',4, '', 'morlet',[-4,4])

9.3.3 添加新小波函数之后

当使用 wavemngr 命令添加一个新小波函数之后，在当前目录中产生 3 个小波扩展文件：两个 ASCII 文件（wavelets.asc 和 wavelets.prv）和 MAT 文件 wavelets.inf。

如果读者想在小波工具箱中使用自己的扩展小波类，必须完成以下步骤：

（1）生成一个新的目录专门存放小波扩展函数；

（2）将前面提到的文件全部移到这个新目录中；

（3）将这个目录利用 Matlab 的 path 命令连接到其搜索路径上；

（4）利用这个目录用于以后的更改；

（5）定义一个 M 文件<fsn>info.m。

这样这个文件就会自动连接到 WaveletFamily 按钮。

第 10 章　小波变换在信号处理中的应用

小波变换作为信号处理的一种手段，逐渐被越来越多领域的理论工作者和工程技术人员所重视和应用，并在许多应用中取得了显著的效果。同传统的处理方法相比，小波变换产生了质的飞跃，在信号处理方面具有更大的优势，比如小波分析可以用于电力负载信号的分析与处理、小波变换可以用于语音信号的分析、变换和综合，还可以检测噪声中未知瞬态信号等。本章将举例说明小波变换工具箱在信号处理某些领域中的应用，希望能抛砖引玉，帮助读者熟练利用小波变换工具箱解决工程实际中遇到的信号处理问题。

本章主要内容包括：

- 小波分解在信号分析中的应用实例；
- 小波变换在信号特征检测中的应用；
- 小波变换在信号降噪和压缩中的应用；
- 小波分析在电力负载信号处理中的应用。

10.1　小波分解在信号分析中的应用实例

本小节以两种典型信号：含噪的三角波正弦波组合信号、含噪的多项式信号为例，主要以介绍如何利用小波分解来分析信号。文中仅对某类小波分析结果做出了一定的分析。有兴趣的读者可以针对上述的各类信号使用小波分析方法，观察所得到的结果，以加深小波分析用于信号处理的基本步骤和过程。

10.1.1　含噪的三角波与正弦波的组合

其表达式为

$$s(t)=\begin{cases}\dfrac{t-1}{500}+\sin(0.3t)+b(t) & 1\le t\le 500\\ \dfrac{1000-t}{500}+\sin(0.3t)+b(t) & 501\le t\le 1000\end{cases}\tag{10.1.1}$$

例程 10-1：应用 db5 小波对式（10.1.1）信号进行 7 层分解。

f10_1.m

```
%生成正弦信号
N=1000;
t=1:N;
sig1=sin(0.3*t);
```

```
%生成三角波信号
sig2(1:500)=((1:500)-1)/500;
sig2(501:N)=(1000-(501:1000))/500;
figure(1);
subplot(2,1,1);
plot(t,sig1,'LineWidth',2);
xlabel('样本序号 n');
ylabel('幅值 A');
subplot(2,1,2);
plot(t,sig2,'LineWidth',2);
xlabel('样本序号 n');
ylabel('幅值 A');
%叠加信号
x=sig1+sig2+randn(1,N);
figure(2)
plot(t,x,'LineWidth',2);
xlabel('样本序号 n');
ylabel('幅值 A');

%一维小波分解
[c,l] = wavedec(x,7,'db5');

%重构第 1-7 层逼近系数.
a7 = wrcoef('a',c,l,'db5',7);
a6 = wrcoef('a',c,l,'db5',6);
a5 = wrcoef('a',c,l,'db5',5);
a4 = wrcoef('a',c,l,'db5',4);
a3 = wrcoef('a',c,l,'db5',3);
a2 = wrcoef('a',c,l,'db5',2);
a1 = wrcoef('a',c,l,'db5',1);

%显示逼近系数
figure(3)
subplot(7,1,1);
plot(a7,'LineWidth',2);
ylabel('a7');
subplot(7,1,2);
plot(a6,'LineWidth',2);
ylabel('a6');
subplot(7,1,3);
plot(a5,'LineWidth',2);
ylabel('a5');
subplot(7,1,4);
```

```
plot(a4,'LineWidth',2);
ylabel('a4');
subplot(7,1,5);
plot(a3,'LineWidth',2);
ylabel('a3');
subplot(7,1,6);
plot(a2,'LineWidth',2);
ylabel('a2');
subplot(7,1,7);
plot(a1,'LineWidth',2);
ylabel('a1');
xlabel('样本序号 n');

%重构第 1-7 层细节系数
d7 = wrcoef('d',c,l,'db5',7);
d6 = wrcoef('d',c,l,'db5',6);
d5 = wrcoef('d',c,l,'db5',5);
d4 = wrcoef('d',c,l,'db5',4);
d3 = wrcoef('d',c,l,'db5',3);
d2 = wrcoef('d',c,l,'db5',2);
d1 = wrcoef('d',c,l,'db5',1);

%显示细节系数
figure(4)
subplot(7,1,1);
plot(d7,'LineWidth',2);
ylabel('d7');
subplot(7,1,2);
plot(d6,'LineWidth',2);
ylabel('d6');
subplot(7,1,3);
plot(d5,'LineWidth',2);
ylabel('d5');
subplot(7,1,4);
plot(d4,'LineWidth',2);
ylabel('d4');
subplot(7,1,5);
plot(d3,'LineWidth',2);
ylabel('d3');
subplot(7,1,6);
plot(d2,'LineWidth',2);
ylabel('d2');
```

```
subplot(7,1,7);
plot(d1,'LineWidth',2);
ylabel('d1');
xlabel('样本序号 n');
```

含噪的三角波与正弦波混合信号波形如图10.1所示。利用db5小波对混合信号进行7层分解，得到的逼近信号如图10.2所示，细节信号如图10.3所示。可以看出：在图10.3中，细节信号d1和d2是与噪声相关的，而d3（特别是d4）是与正弦信号相关的；而在图10.2中，逼近信号d7是一个三角波。

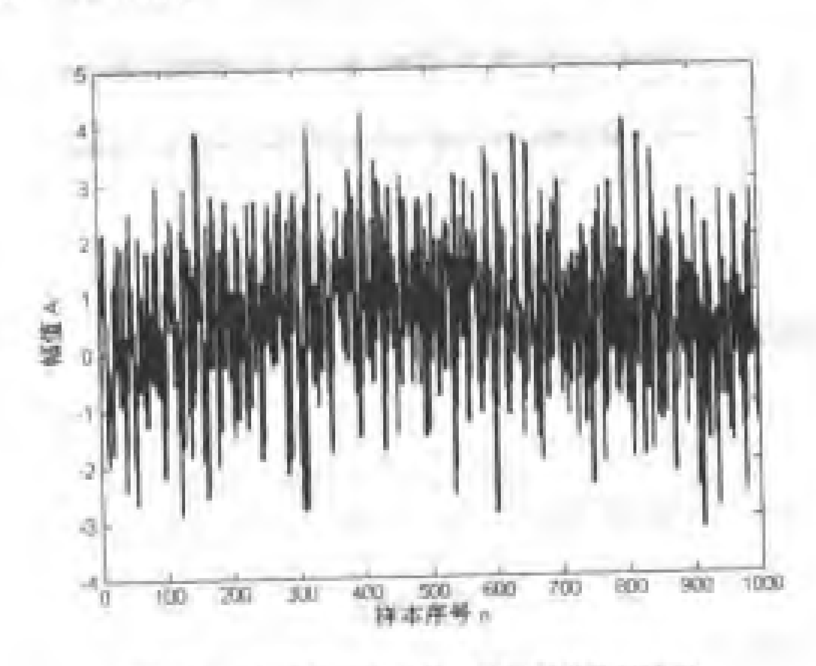


图10.1 含噪的三角波与正弦波混合信号波形

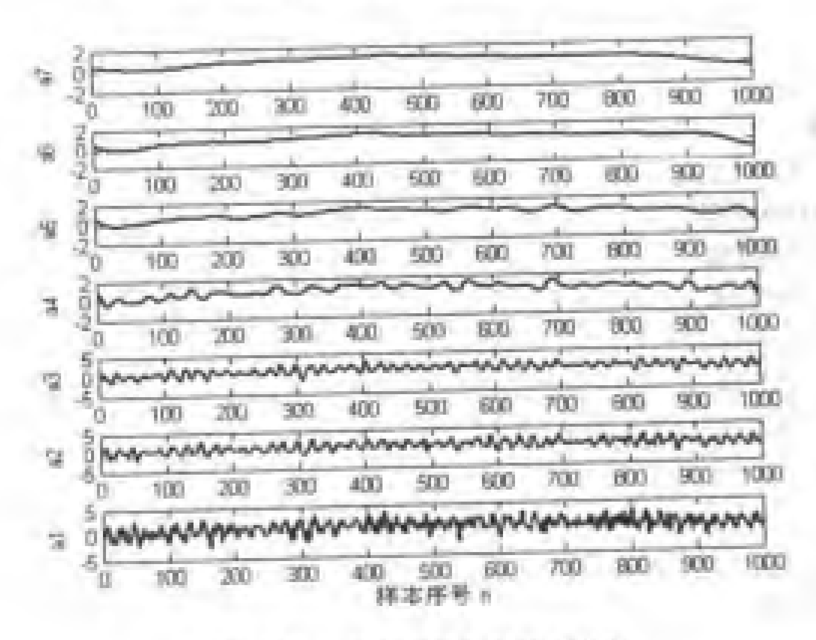


图10.2 小波分解后各层逼近信号

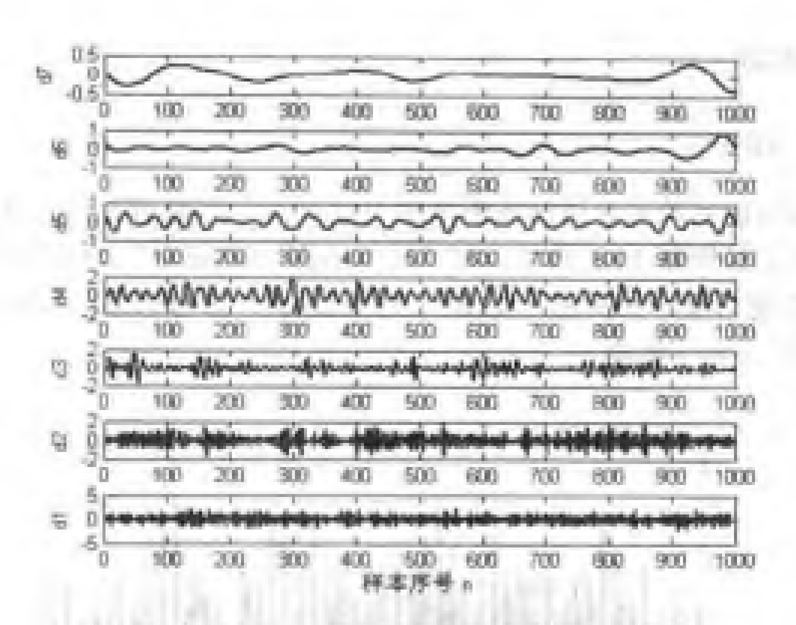


图 10.3　小波分解后各层细节信号

10.1.2　含噪的多项式信号

其表达式为

$$s(t)=t^2-t+1+b_1(t) \tag{10.1.2}$$

例程 10-2：分别应用 db2 和 db3 小波对式（10.1.2）信号进行 4 层分解。

f10_2.m

```
%生成含噪的多项式信号
N=800;
t=1:N;
sig=t.^2-t+1;
x=sig+randn(1,N);
%一维小波分解
[c,l] = wavedec(x,4,'db3');
%[c,l] = wavedec(x,4,'db2');
%重构第 1-4 层逼近系数.
a4 = wrcoef('a',c,l,'db3',4);
a3 = wrcoef('a',c,l,'db3',3);
a2 = wrcoef('a',c,l,'db3',2);
a1 = wrcoef('a',c,l,'db3',1);
%显示逼近系数
figure(1)
subplot(4,1,1);
plot(a4,'LineWidth',2);
ylabel('a4');
subplot(4,1,2);
```

```
plot(a3,'LineWidth',2);
ylabel('a3');
subplot(4,1,3);
plot(a2,'LineWidth',2);
ylabel('a2');
subplot(4,1,4);
plot(a1,'LineWidth',2);
ylabel('a1');
xlabel('样本序号 n');
%重构第 1-4 层细节系数
d4 = wrcoef('d',c,l,'db3',4);
d3 = wrcoef('d',c,l,'db3',3);
d2 = wrcoef('d',c,l,'db3',2);
d1 = wrcoef('d',c,l,'db3',1);
%显示细节系数
figure(2)
subplot(4,1,1);
plot(d4,'LineWidth',2);
ylabel('d4');
axis([0 N -100 100]);
subplot(4,1,2);
plot(d3,'LineWidth',2);
ylabel('d3');
axis([0 N -30 30]);
subplot(4,1,3);
plot(d2,'LineWidth',2);
ylabel('d2');
axis([0 N -5 5]);
subplot(4,1,4);
plot(d1,'LineWidth',2);
ylabel('d1');
xlabel('样本序号 n');
```

利用 db2 小波分解后的逼近信号如图 10.4 所示，细节信号如图 10.5 所示。可以看出：这种情况下随着分解层级的增加，其正则性增加，从而抑制了该多项式信号的零阶和一阶部分，而仅对该信号的二阶部分以及噪声进行了分解，因此在图 10.5 中，除了细节信号 d1 中包含了该含噪信号的不规则性，其余各层细节中的信号周期性（规则性）随着层级的增加而增大。

利用 db3 小波分解后的逼近信号如图 10.6 所示，细节信号如图 10.7 所示。可以看出：由于 db3 小波的正则性较差，所以它抑制了该信号的多项式部分，而析出了它的噪声部分，因此利用小波分析可以较好地对该类信号进行抑制。

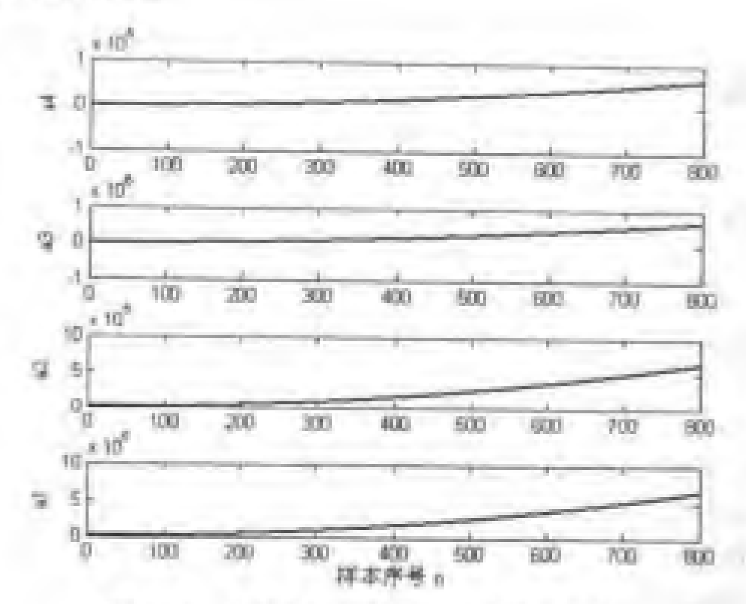


图 10.4 小波分解后各层逼近信号（db2）

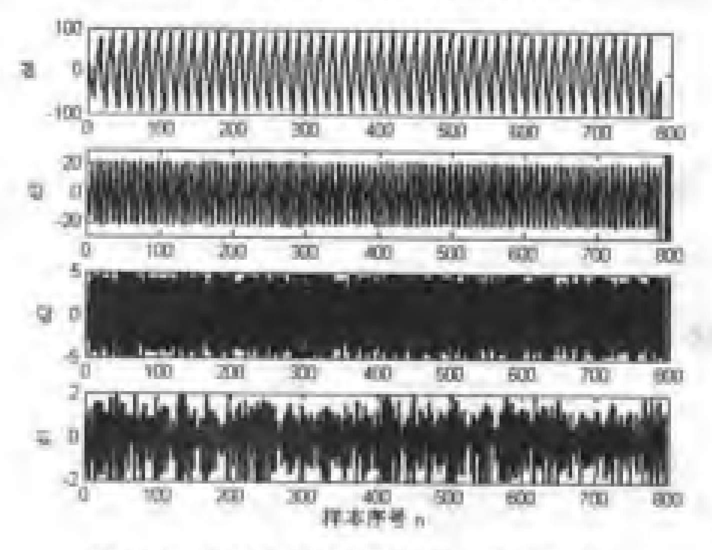


图 10.5 小波分解后各层细节信号（db2）

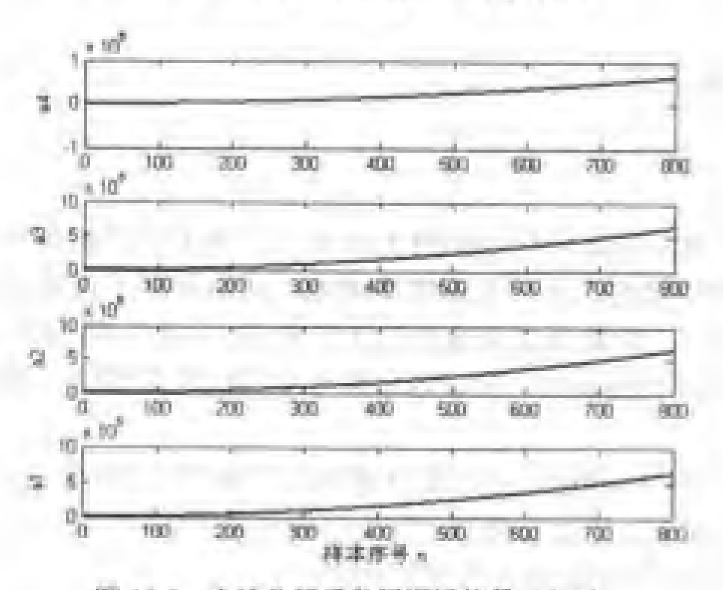


图 10.6 小波分解后各层逼近信号（db3）

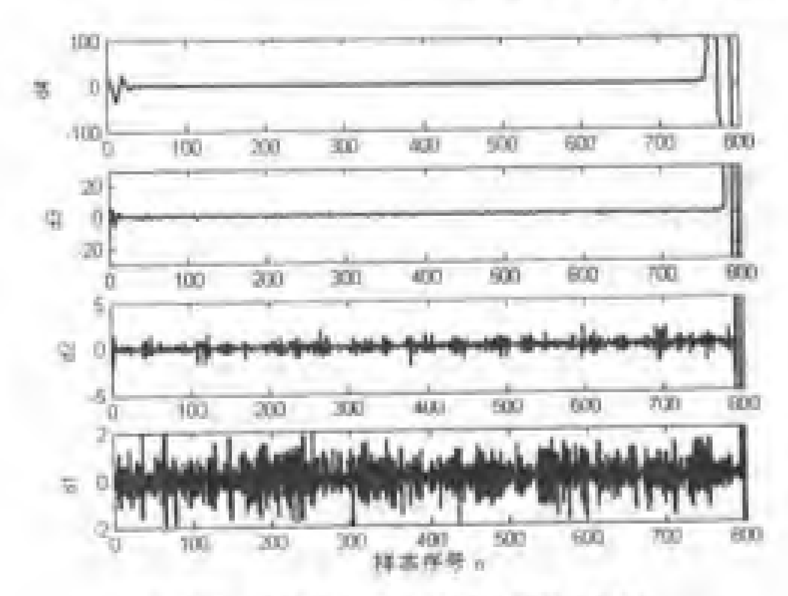


图 10.7　小波分解后各层细节信号（db3）

10.2　小波变换在信号降噪和压缩中的应用

10.2.1　信号降噪

小波分析的重要应用之一就是用于信号消噪，首先简要地阐述一下小波变换实现信号消噪的基本原理。

含噪的一维信号模型可以表示如下：

$$s(k)=f(k)+\varepsilon\cdot e(k),\ k=0,1,\cdots,n-1$$

式中 $s(k)$ 为含噪信号，$f(k)$ 为有用信号，$e(k)$ 为噪声信号。这里假设 $e(k)$ 是一个高斯白噪声，通常表现为高频信号，而工程实际中 $f(k)$ 通常为低频信号或者是一些比较平稳的信号。因此我们可按如下方法进行消噪处理：首先对信号进行小波分解，由于噪声信号多包含在具有较高频率的细节中，从而可以利用门限、阈值等形式对分解所得的小波系数进行处理，然后对信号进行小波重构即可达到对信号消噪的目的。

对信号消噪实质上是抑制信号中的无用部分，增强信号中有用部分的过程。一般地，一维信号消噪的过程可分为如下 3 个步骤。

步骤 1：一维信号的小波分解。选择一个小波并确定分解的层次，然后进行分解计算。

步骤 2：小波分解高频系数的阈值量化。对各个分解尺度下的高频系数选择一个阈值进行软阈值量化处理。

步骤 3：一维小波重构。根据小波分解的最底层低频系数和各层高频系数进行一维小波重构。

这 3 个步骤中，最关键的是如何选择阈值以及进行阈值量化。在某种程度上，它关系到信号消噪的质量。

1．噪声在小波分解下的特性

总体上，对于一维离散信号来说，其高频部分所影响的是小波分解的第一层细节，其低

频部分所影响的是小波分解的最深层和低频层。如果对一个仅由白噪声所组成的信号进行分析，则可得出这样的结论：高频系数的幅值随着分解层次的增加而迅速地衰减，且其方差也有同样的变化趋势。

用 $C(j,k)$表示噪声经过小波分解后的系数，其中 j 表示尺度，k 表示时间。下面将噪声看作普通信号，分析它的相关性、频谱和频率分布这 3 个主要特征。

（1）如果所分析的信号 s 是一个平稳、零均值的白噪声，那么它的小波分解系数是相互独立的。

（2）如果信号 s 是一个高斯型噪声，那么其小波分解系数是互不相关的，且服从高斯分布。

（3）如果信号 s 是一个平稳、有色、零均值的高斯型噪声序列，那么它的小波分解系数也是高斯序列，并且对每一个分解尺度 j，其相应的系数也是一个平稳、有色的序列。如何选择对分解系数具有解相关性的小波是一个很困难的问题，在目前也没有得到很好地解决。进一步需要指出的是，即使存在这样的一个小波，但它对噪声的解相关性还取决于噪声的有色性。为了利用小波计算噪声的解相关性，必须事先知道噪声本身的特性。

（4）如果信号 s 是一个固定的、零均值 ARMA 模型，那么对每一个小波分解尺度 j，$C(j,k)\ (k\in z)$也是固定的、零均值 ARMA 模型，且其特性取决于尺度 j。

（5）如果信号 s 是一个一般的噪声：

① 若它的相关函数已知，则可计算系数序列 $C(j,k)$和 $C(j,k')$；

② 若它的相关函数谱已知，则可计算 $C(j,k)(k\in z)$的谱及尺度 j 和 j'的交叉谱。

2. 应用一维小波分析进行信号消噪处理

小波分析工具箱中用于信号消噪的一维小波函数是 wden.m 和 wdencmp.m。

小波分析进行消噪处理一般有下述 3 种方法。

（1）默认阈值消噪处理。该方法利用函数 ddencmp 生成信号的默认阈值，然后利用函数 wdencmp 进行消噪处理。

（2）给定阈值消噪处理。在实际的消噪处理过程中，阈值往往可通过经验公式获得，且这种阈值比默认阈值的可信度高。在进行阈值量化处理时可利用函数 wthresh。

（3）强制消噪处理。该方法是将小波分解结构中的高频系数全部置为 0，即滤掉所有高频部分，然后对信号进行小波重构。这种方法比较简单，且消噪后的信号比较平滑，但是容易丢失信号中的有用成份。

例程 10-3：利用小波分析对含噪正弦波进行消噪。

f10_3.m

```
%生成正弦信号
N=1000;
t=1:N;
x=sin(0.03*t);
%加噪声
load noissin;
ns=noissin;
%显示波形
```

```
subplot(3,1,1);
plot(t,x);
xlabel('样本序号 n');
ylabel('(原始信号)幅值 A');
subplot(3,1,2);
plot(ns);
xlabel('样本序号 n');
ylabel('(含噪信号)幅值 A');
%小波消噪
xd=wden(ns,'minimaxi','s','one',5,'db3');
subplot(3,1,3);
plot(xd);
xlabel('样本序号 n');
ylabel('(消噪信号)幅值 A');
```

信号消噪结果如图 10.8 所示。

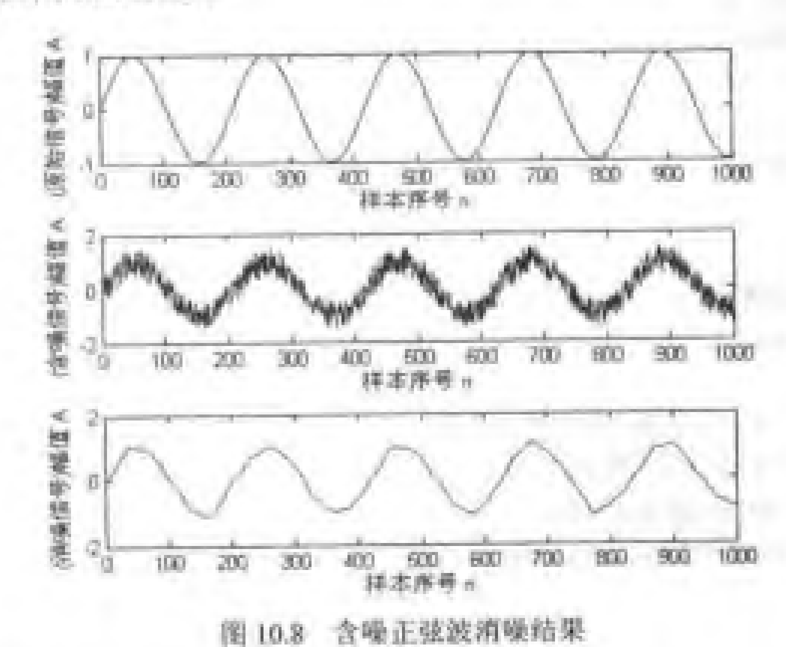


图 10.8　含噪正弦波消噪结果

从图 10.8 中可以看出：消噪后的信号大体上恢复了原始信号的形状，并明显地除去了噪声所引起的干扰。但是，恢复后的信号和原始信号相比，有明显的改变。这主要是因为在进行消噪处理的过程中所用的分析小波和细节系数阈值不恰当所致。下面我们再通过一个例程对此进行说明。

例程 10-4：在电网电压值监测过程中，由于监测设备出现了一点故障，致使所采集到的信号受到噪声的污染。现在利用小波分析对污染信号进行消噪处理以恢复原始信号。

f10_4.m

```
%装载采集的信号 leleccum.mat
load leleccum;
```

```
%将信号中第 2000 到第 3450 个采样点赋给 s
indx=2000:3450;
s=leleccum(indx);
%画出原始信号
subplot(2,2,1);
plot(s);
title('原始信号');
ylabel('幅值 A');
%用 db1 小波对原始信号进行 3 层分解并提取系数
[c,l]=wavedec(s,3,'db1');
a3=appcoef(c,l,'db1',3);
d3=detcoef(c,l,3);
d2=detcoef(c,l,2);
d1=detcoef(c,l,1);
%对信号进行强制性消噪处理并图示结果
dd3=zeros(1,length(d3));
dd2=zeros(1,length(d2));
dd1=zeros(1,length(d1));
c1=[a3 dd3 dd2 dd1];
s1=waverec(c1,l,'db1');
subplot(2,2,2);
plot(s1);
title('强制消噪后的信号');
xlabel('样本序号 n');
ylabel('幅值 A');
%用默认阈值对信号进行消噪处理并图示结果
%用 ddencmp 函数获得信号的默认阈值
[thr,sorh,keepapp]=ddencmp('den','wv',s);
s2=wdencmp('gbl',c,l,'db1',3,thr,sorh,keepapp);
subplot(2,2,3);
plot(s2);
title('默认阈值消噪后的信号');
xlabel('样本序号 n');
ylabel('幅值 A');
%用给定的软阈值进行消噪处理
softd1=wthresh(d1,'s',1.465);
softd2=wthresh(d2,'s',1.823);
softd3=wthresh(d3,'s',2.768);
c2=[a3 softd3 softd2 softd1];
s3=waverec(c2,l,'db1');
subplot(2,2,4);
plot(s3);
title('给定软阈值消噪后的信号');
```

```
xlabel('样本序号 n');
ylabel('幅值 A');
```

计算结果如图 10.9 所示。

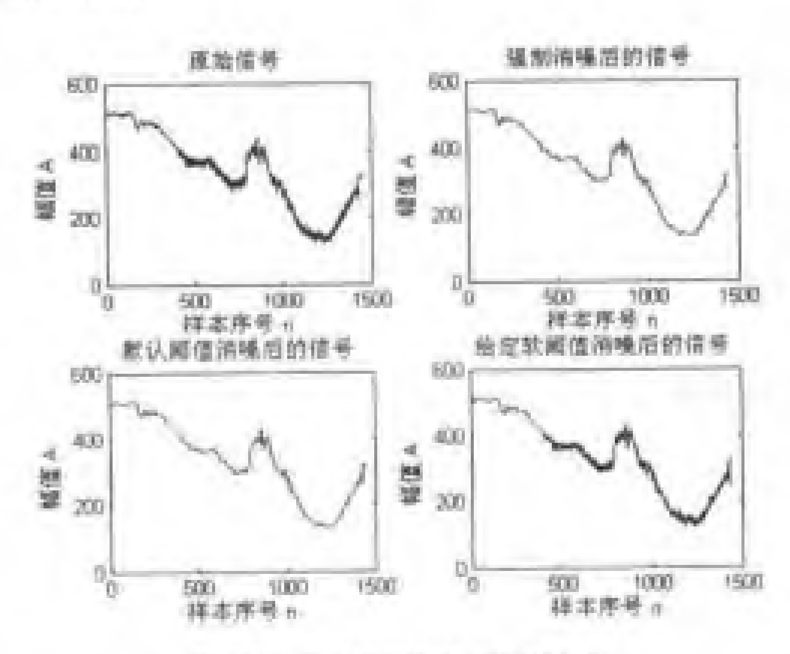


图 10.9　3 种不同消噪方法的结果

在本例程中，分别利用前面所提到的 3 种消噪方法进行处理。从图 10.9 得到的结果来看：应用强制消噪处理后的信号较为光滑，但是它很有可能丢失了信号中的一些有用成分；默认阈值消噪和给定软阈值消噪这两种处理方法在实际中应用得更为广泛一些。

3. 消噪阈值选取规则

在 Matlab 的小波工具箱中，设置软或硬阈值的函数为 wthresh.m。

该函数根据参数 sorh 的值计算分解系数的软阈值或硬阈值。其中，硬阈值对应于最简单的处理方法，而软阈值具有很好的数学特性，并且所得到的理论结果是可用的。

例程 10-5：生成不同阈值下的信号。

f10_5.m

```
%生成线性信号
y=linspace(-1,1,100);
%设置 T 阈值
thr=0.4;
%计算软、硬阈值
ythard=wthresh(y,'h',thr);
ytsoft=wthresh(y,'s',thr);
%显示不同阈值后的信号
subplot(1,3,1);
plot(y);
title('原始信号');
```

```
xlabel('样本序号 n');
ylabel('幅值 A');
subplot(1,3,2);
plot(ythard);
title('硬阈值信号');
xlabel('样本序号 n');
ylabel('幅值 A');
subplot(1,3,3);
plot(ytsoft);
title('软阈值信号');
xlabel('样本序号 n');
ylabel('幅值 A');
```

计算结果如图 10.10 所示。

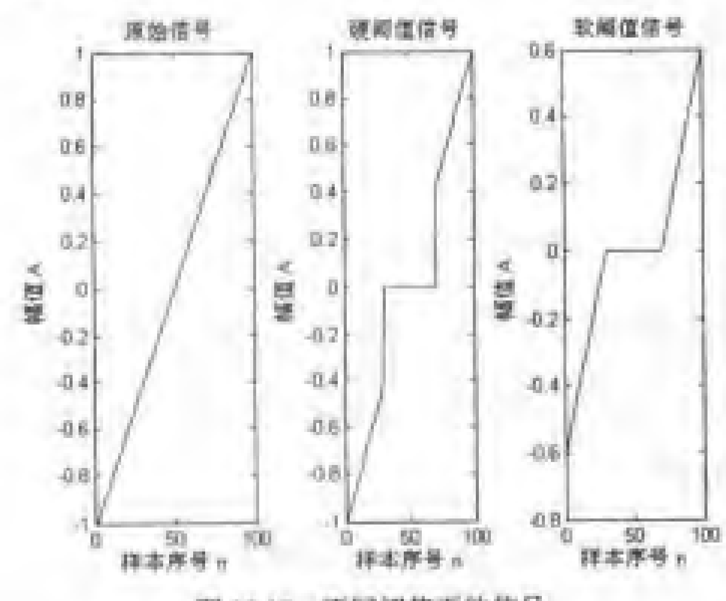


图 10.10　不同阈值下的信号

说明：令 t 表示阈值，则硬阈值信号 s 的形式为：$s=\begin{cases} x & |x|>t \\ 0 & |x|\le t \end{cases}$

软阈值信号 s 的形式为：$s=\begin{cases} sign(x)(|x|-t) & |x|>t \\ 0 & |x|\le t \end{cases}$。

根据基本的噪声模型，阈值的选取有以下 4 个规则，其中每一条规则对应于函数 thselect 中输入参数 tptr 的一个选项。

（1）选项 tptr='rigrsure'，是一种基于 Stein 无偏似然估计原理的自适应阈值选择。给定一个阈值 t，得到它的似然估计，再将非似然最小化，就可得到所选的阈值。这是一种软件阈值估计器。

（2）选项 tptr='sqtwolog'，是一种固定的阈值形式，它所产生的阈值为 sprt(2*log(length(X)))。

（3）选项 tptr='heursure'，是前两种阈值的综合，所选择的是最优预测变量阈值。如果信

噪比很小，而SURE估计有很大的噪声，此时就需采用这种固定的阈值形式。

（4）选项tptr='minimaxi'，也是一种固定的阈值选择形式，它所产生的是一个最小均方差的极值，而不是无误差。

例程10-6：计算高斯白噪声信号$N(0,1)$在上述4条规则下的阈值。

f10_6.m

```
%产生白噪声
y=randn(1,1000);
%生成不同的阈值
thr1=thselect(y,'rigrsure')
thr2=thselect(y,'sqtwolog')
thr3=thselect(y,'heursure')
thr4=thselect(y,'minimaxi')
```

计算结果为：

```
thr1 =
    2.7316
thr2 =
    3.7169
thr3 =
    3.7169
thr4 =
    2.2163
```

在本例程中，信号y是一个标准高斯白噪声，所以每一种方法应该都能粗略地将所有系数剔除。从计算的结果来看，对于Stein的无偏似然估计（SURE）和极大极小（minimaxi）原理的阈值选择规则，仅约保存了3%的系数；而其他两种阈值选择规则，将所有的系数都变成了零。

同样地，对噪声进行小波分解时，也会产生高频系数，故一个信号的高频系数向量是有用信号和噪声信号的高频系数的叠加。由于SURE和minimaxi阈值选取规则较为保守（仅将部分系数置为零），因此在信号的高频信息有很少一部分在噪声范围内时，这两种阈值非常有用，可以将弱小的信号提取出来。其他两种阈值选取规则，在去除噪声时更为有效，但是也可能将有用信号的高频部分当作噪声信号去除掉。

在实际的工程应用中，大多数信号可能包含着许多尖峰或突变，而且噪声信号也并不是平稳的白噪声。对这种信号进行消噪处理时，传统的傅立叶变换完全是在频域中对信号进行分析，它不能给出信号在某个时间点上的变化情况，因此分辨不出信号在时间轴上的任何一个突变。但是小波分析能同时在时频域内对信号进行分析，所以它能有效地区分信号中的突变部分和噪声，从而实现对非平稳信号的消噪。

下面通过一个实例，考察小波分析对非平稳信号的消噪。

例程10-7：利用小波分析对一个含噪的矩形波信号进行消噪处理。

f10_7.m

```
%设置信噪比和随机种子值
snr=4;
```

```
init=2055615866;
%产生原始信号 sref 和被高斯白噪声污染的信号 s
[sref,s]=wnoise(1,11,snr,init);

%用 sym8 小波对信号 s 进行 3 层分解并对细节系数
%选用 sure 阈值模式和尺度噪声
xd=wden(s,'heursure','s','one',3,'sym8');

%显示信号波形
subplot(3,1,1);
plot(sref);
xlabel('样本序号 n');
ylabel('幅值 A');
title('参考信号');
subplot(3,1,2);
plot(s);
title('含噪信号');
xlabel('样本序号 n');
ylabel('幅值 A');
subplot(3,1,3);
plot(xd);
title('消噪信号');
xlabel('样本序号 n');
ylabel('幅值 A');
```

计算结果如图 10.11 所示。

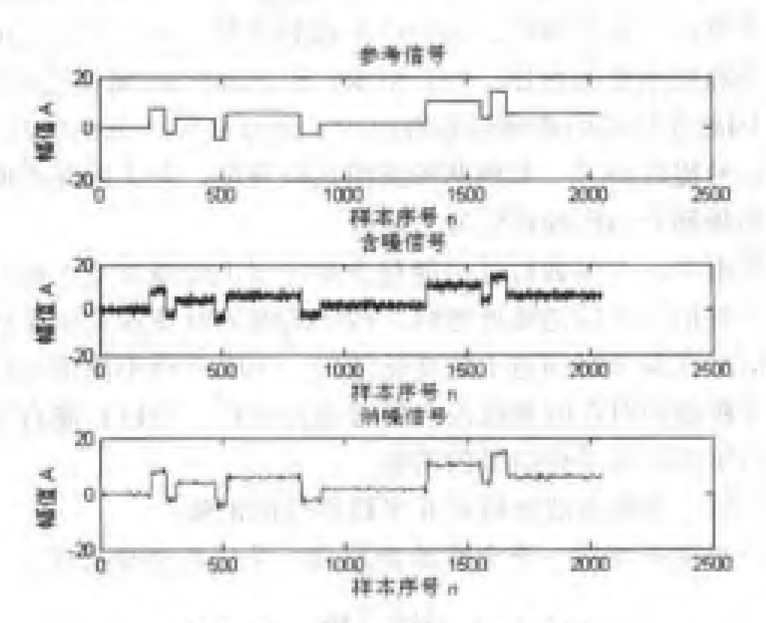


图 10.11 含噪矩形波的消噪结果

10.2.2 信号压缩

应用一维小波分析之所以能对信号进行压缩，是因为一个比较规则的信号是由一个数据量很小的低频系数和几个高频层的系数所组成的。这里对低频系数的选择有一个要求，即需要在一个合适的分解层上选取低频系数。对一维信号进行压缩，可以选用小波分析和小波包分析两种手段进行，主要包括以下几个步骤。

步骤1：信号的小波（或小波包）分解。

步骤2：对高频系数进行阈值量化处理。对第1到第N层的高频系数，均可选择不同的阈值，并且用硬阈值进行系数的量化。

步骤3：对量化后的系数进行小波（或小波包）重构。

一般地，有两种比较有效的信号压缩方法，第一种方法是对信号进行小波尺度的扩展，并且保留绝对值最大的系数。在这种情况下，可以选择使用全局阈值，此时仅需要输入一个参数即可。第二种方法是根据分解后各层的效果来确定某一层的阈值，且每一层的阈值可以是互不相同的。

下面给出一个具体的例程，以便使读者对小波分析在信号压缩中的应用有一个较为直观的印象。

例程10-8：利用小波分析对给定信号进行压缩处理。

f10_8.m

```
%装载信号
load leleccum;
%截取信号中的一段[2600，3100]
s=leleccum(2600:3100);
%用小波db3对s进行三层分解
[c,l]=wavedec(s,3,'db3');
%选用全局阈值进行信号压缩处理
thr=40;
[sd,csd,lsd,perf0,perfl2]=wdencmp('gbl',c,l,'db3',3,thr,'h',1);
subplot(2,1,1);
plot(s,'LineWidth',2);
title('原始信号');
xlabel('样本序号 n');
ylabel('幅值 A');
subplot(2,1,2);
plot(sd,'LineWidth',2);
title('压缩后的信号');
xlabel('样本序号 n');
ylabel('幅值 A');
```

计算结果如图10.12所示。

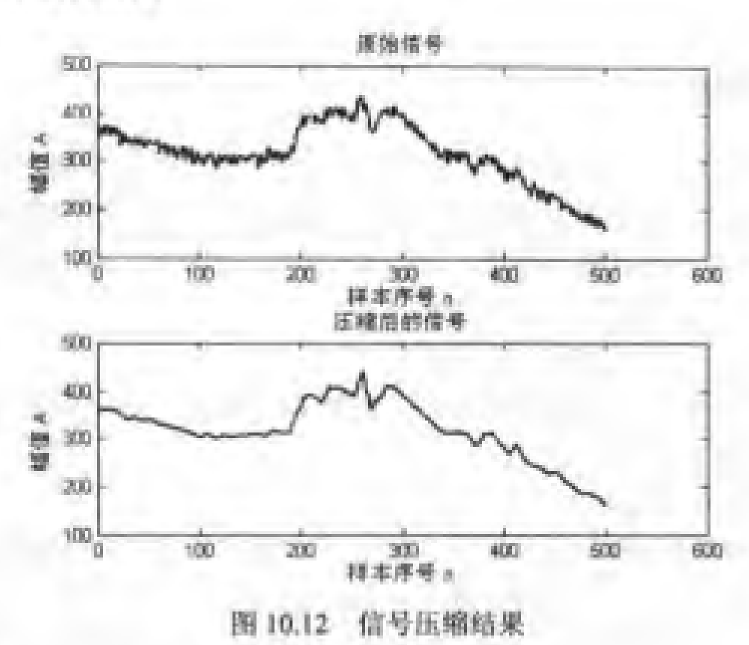


图 10.12　信号压缩结果

注意：信号的压缩与消噪相比，主要差别在第二步。

10.3　小波分析在电力负载信号处理中的应用

电力系统在线监测中的监测信号含有大量的现场背景噪声，给传统方式的数据采集与故障诊断带来很大的困难，将以处理瞬态信号、含宽带噪声信号等见长的小波分析应用于电力系统在线监测是大有前途的。

本小节的测量数据是从一个复杂的设备上采集的电力负载信号，每分钟采集一个样本，持续了 5 个星期，总共 50 400 个数据样本。测量数据受到传感器误差和状态噪声两种噪声的影响。本小节将分析其中的两段数据，其中第一段是上午 12:30 至下午 1:00 间采集的样本，由于这段时间处于用电高峰，因此数据很复杂；第二段是下半夜采集的样本，数据比较简单。

10.3.1　信号分解

例程 10-9：利用小波分解分析第一段数据的信号成分。

f10_9.m

```
%装载采集的信号 leleccum.mat
load leleccum;
%将信号中第 3600 到第 3700 个采样点赋给 s
index=3600:3700;
s=leleccum(index);
%画出原始信号
figure(1);
plot(index,s);
ylabel('幅值 A');
```

```
xlabel('样本序号 n');
%用 db3 小波进行 5 层分解
[c,l]=wavedec(s,5,'db3');
%重构第 1-5 层逼近系数.
a5 = wrcoef('a',c,l,'db3',5);
a4 = wrcoef('a',c,l,'db3',4);
a3 = wrcoef('a',c,l,'db3',3);
a2 = wrcoef('a',c,l,'db3',2);
a1 = wrcoef('a',c,l,'db3',1);
%显示逼近系数
figure(2)
subplot(5,2,1);
plot(index,a5,'LineWidth',2);
ylabel('a5');
subplot(5,2,3);
plot(index,a4,'LineWidth',2);
ylabel('a4');
subplot(5,2,5);
plot(index,a3,'LineWidth',2);
ylabel('a3');
subplot(5,2,7);
plot(index,a2,'LineWidth',2);
ylabel('a2');
subplot(5,2,9);
plot(index,a1,'LineWidth',2);
ylabel('a1');
xlabel('样本序号 n');
%重构第 1-5 层细节系数
d5 = wrcoef('d',c,l,'db3',5);
d4 = wrcoef('d',c,l,'db3',4);
d3 = wrcoef('d',c,l,'db3',3);
d2 = wrcoef('d',c,l,'db3',2);
d1 = wrcoef('d',c,l,'db3',1);
%显示细节系数
subplot(5,2,2);
plot(index,d5,'LineWidth',2);
ylabel('d5');
subplot(5,2,4);
plot(index,d4,'LineWidth',2);
ylabel('d4');
subplot(5,2,6);
plot(index,d3,'LineWidth',2);
ylabel('d3');
```

```
subplot(5,2,8);
plot(index,d2,'LineWidth',2);
ylabel('d2');
subplot(5,2,10);
plot(index,d1,'LineWidth',2);
ylabel('d1');
xlabel('样本序号 n');
```

第一段电力载波信号如图 10.13 所示，利用 db3 小波对其进行 5 层分解，得到的逼近信号和细节信号如图 10.14 所示。可以看出：细节信号 d1 和 d2 的值较小，可以认为是由传感器和状态噪声的高频分量引起的局部的干扰；细节信号 d4 包含了 3 个相连的主要信号模式，它最接近于原始数据的曲线；而细节信号 d5 含有的信息不多，因此第 4 层贡献最大，它提取了原始数据曲线的形状。

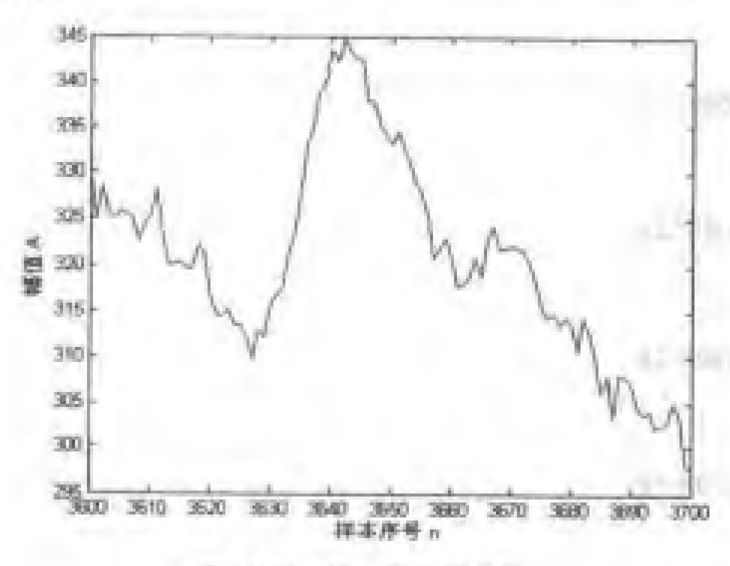


图 10.13　第一段数据波形

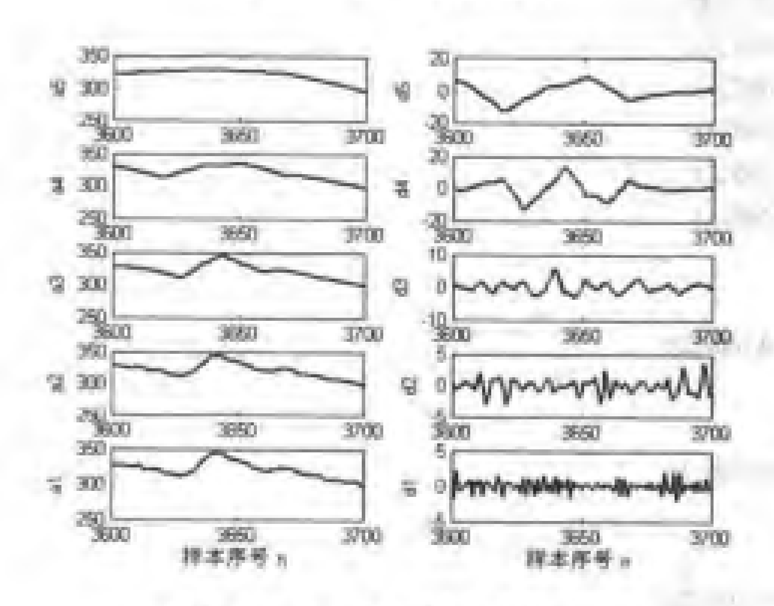


图 10.14　第一段数据小波分解结果

10.3.2 暂态信号检测

为保证电力系统的安全可靠运行，必须对电力设备进行状态监测，根据电力信号来判别其运行的状态。电力系统暂态故障信号往往在故障时刻发生突变，若能捕获设备故障信息突变时刻和大小，有利于在故障初期及早采取措施使系统恢复正常，这对提高设备运行可靠性具有重要的意义。

例程 10-10：利用小波分析检测第二段信号的突变点。

f10_10.m

```
%装载采集的信号 leleccum.mat
load leleccum;
%将信号中第 3600 到第 3700 个采样点赋给 s
index=1575:1720;
s=leleccum(index);
%画出原始信号
figure(1);
plot(index,s);
xlabel('样本序号 n');
ylabel('幅值 A');
%用 db3 小波进行 5 层分解
[c,l]=wavedec(s,5,'db3');
……%以下同上
```

第二段电力载波信号如图 10.15 所示，该段数据样本在时间 t=1 600 和 t=1 625 两处存在突变点。

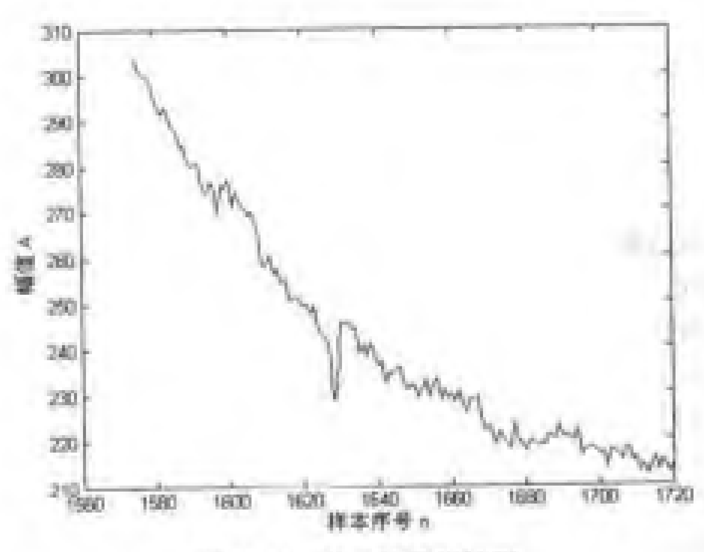


图 10.15　第二段数据波形

利用 db3 小波对其进行 5 层分解，得到的逼近信号和细节信号如图 10.16 所示。可以看出：由细节信号 d2 可以检测突变点位置 t=1 625，由细节信号 d1 也能隐约看出 t=1 600 处的突变点。

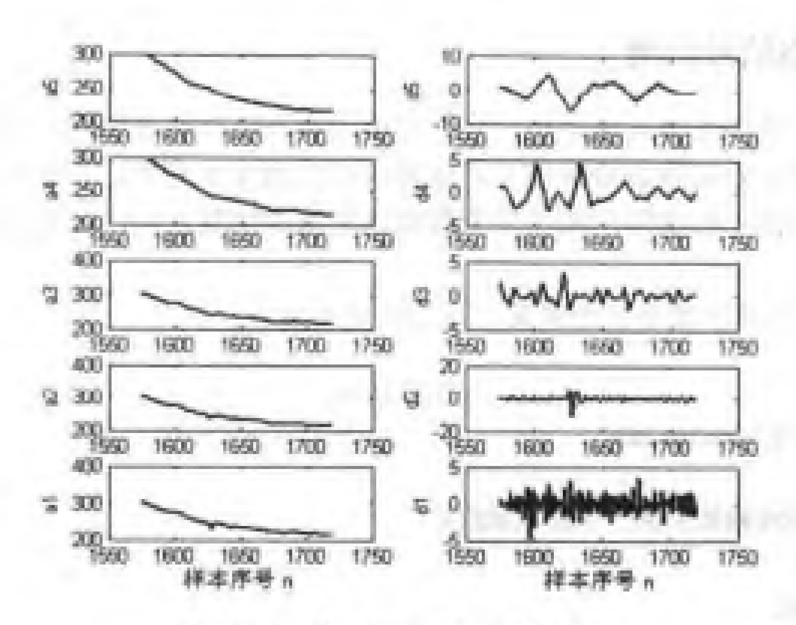


图 10.16　第二段数据小波分析结果

10.3.3　传感器故障检测

例程 10-11：利用小波分析检测传感器故障。

f10_11.m

```
%装载采集的信号 leleccum.mat
load leleccum;
%将信号中第 2200 到第 3600 个采样点赋给 s
index=2200:3600;
s=leleccum(index);
%画出原始信号
figure(1);
plot(index,s);
xlabel('样本序号 n');
ylabel('幅值 A');
%用 db3 小波进行 5 层分解
[c,l]=wavedec(s,5,'db3');
%重构第 1-5 层细节系数
d5 = wrcoef('d',c,l,'db3',5);
d4 = wrcoef('d',c,l,'db3',4);
d3 = wrcoef('d',c,l,'db3',3);
d2 = wrcoef('d',c,l,'db3',2);
d1 = wrcoef('d',c,l,'db3',1);
%显示细节系数
figure(2)
subplot(3,1,1);
plot(index,d3,'LineWidth',2);
```

```
ylabel('d3');
subplot(3,1,2);
plot(index,d2,'LineWidth',2);
ylabel('d2');
subplot(3,1,3);
plot(index,d1,'LineWidth',2);
ylabel('d1');
xlabel('样本序号 n');
```

采集的电力负载信号波形如图 10.17 所示，从图中可以看出在两个黑圆圈之间的信号出现异常，这是由于传感器故障造成的，为了更清楚地揭示这种故障，利用 db3 小波对其进行 5 层分解，得到的第 1～3 层的细节信号如图 10.18 所示。可以看出：每个细节信号都显示了在 t=2 400 和 t=3 600 之间的信号由于传感器故障而引入了传感器误差噪声。

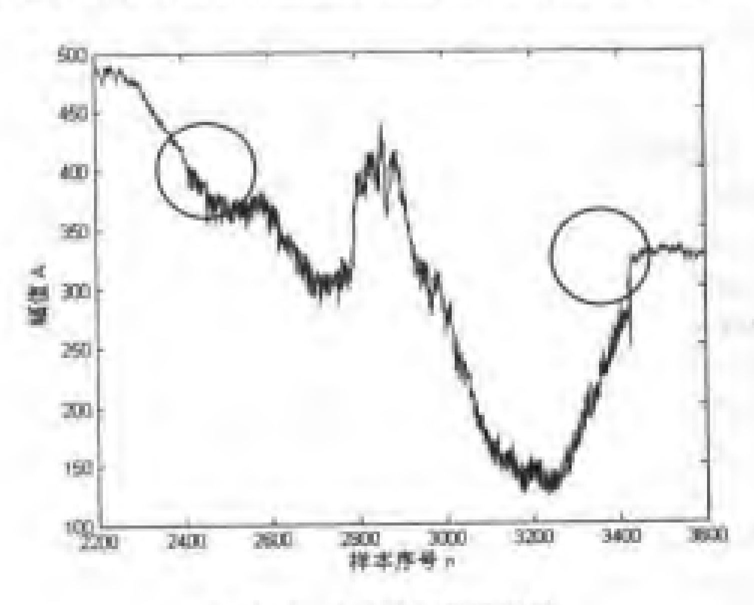


图 10.17　电力负载信号波形

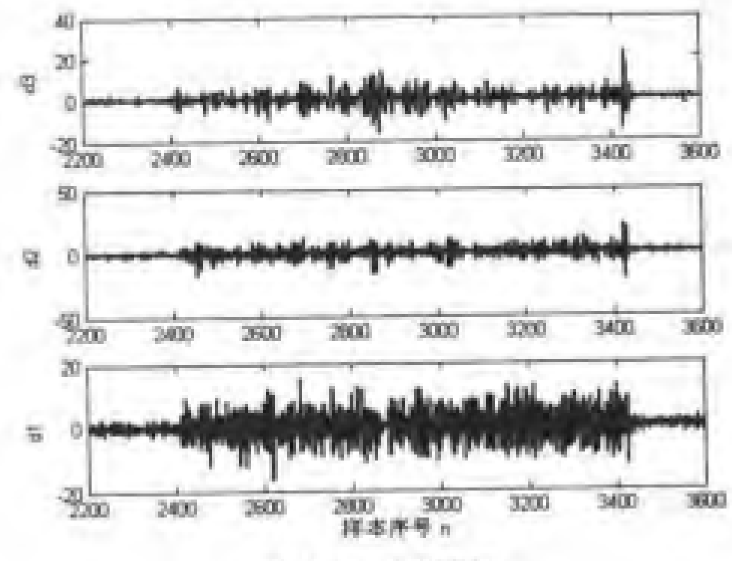


图 10.18　小波分解

10.3.4 奇异点定位和消除

例程 10-12：利用小波分析检测信号中的奇异点并消除。

f10_12.m

```
%装载采集的信号 leleccum.mat
load leleccum;
%将信号中第 1160 到第 1235 个采样点赋给 s
index=1160:1235;
s=leleccum(index);
%画出原始信号
figure(1);
plot(index,s);
xlabel('样本序号 n');
ylabel('幅值 A');

%用 db3 小波进行 5 层分解
[c,l]=wavedec(s,5,'db3');
%重构第 1-5 层逼近信号
a5 = wrcoef('a',c,l,'db3',5);
a4 = wrcoef('a',c,l,'db3',4);
a3 = wrcoef('a',c,l,'db3',3);
a2 = wrcoef('a',c,l,'db3',2);
a1 = wrcoef('a',c,l,'db3',1);
%重构第 1-5 层细节信号
d5 = wrcoef('d',c,l,'db3',5);
d4 = wrcoef('d',c,l,'db3',4);
d3 = wrcoef('d',c,l,'db3',3);
d2 = wrcoef('d',c,l,'db3',2);
d1 = wrcoef('d',c,l,'db3',1);

%显示细节信号
figure(2)
subplot(3,1,1);
plot(index,d3,'LineWidth',2);
ylabel('d3');
subplot(3,1,2);
plot(index,d2,'LineWidth',2);
ylabel('d2');
subplot(3,1,3);
plot(index,d1,'LineWidth',2);
ylabel('d1');
xlabel('样本序号 n');
```

```
%消除奇异点
%%设置 1-3 层细节信号为零
s0=a5+d5+d4;
%画出重构信号
figure(3);
plot(index,s0);
xlabel('样本序号 n');
ylabel('幅值 A');
```

采集的原始信号波形如图 10.19 所示，可以明显看见在 t=1 193 和 t=1 215 两处存在奇异值点。进一步利用 db3 小波对信号进行 5 层分解，得到的 1～3 层细节信号如图 10.20 所示，发现奇异值点包含在细节信号 d1 和 d2，且与原信号中奇异点是同步的。为了消除奇异点，重构信号时令细节信号 d1、d2 和 d3 等于零，得到的信号波形如图 10.21 所示，比较图 10.19 和图 10.21 可见，奇异值点已经很不明显了。

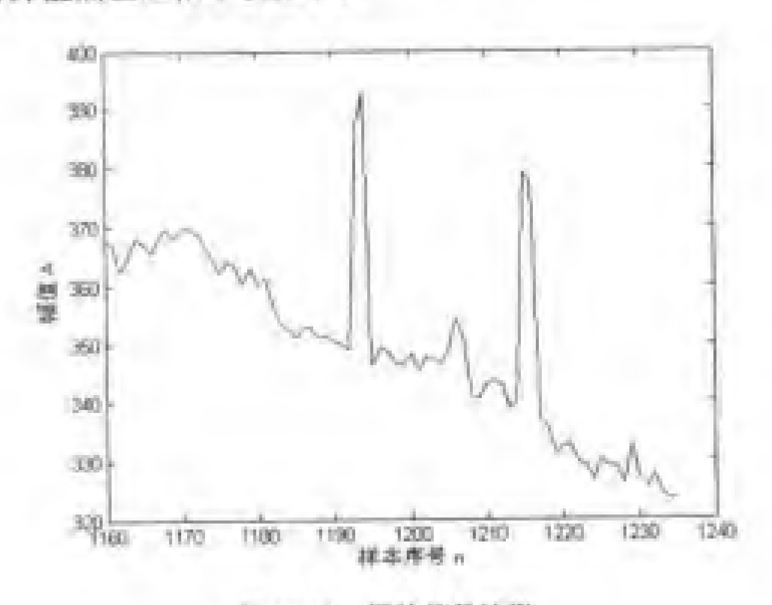


图 10.19 原始信号波形

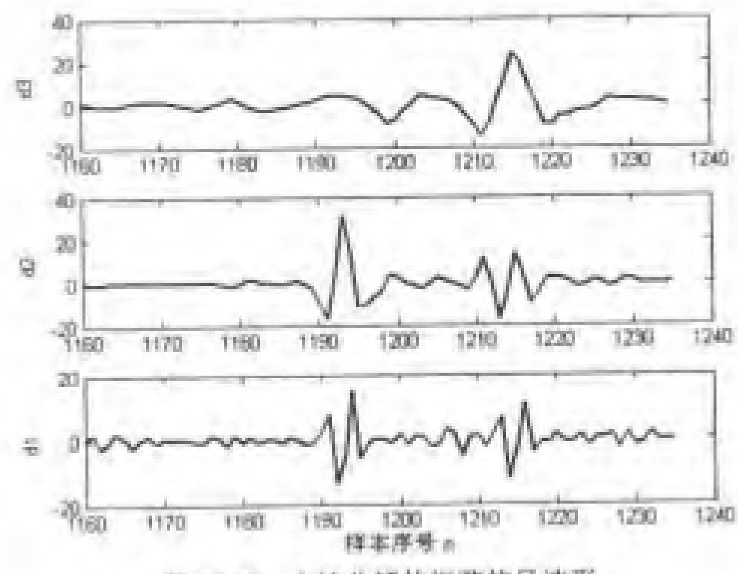


图 10.20 小波分解的细节信号波形

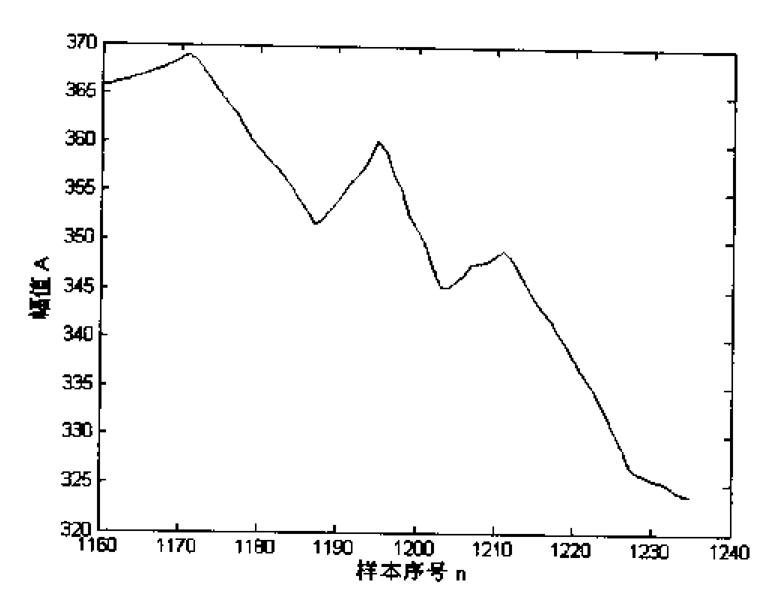


图 10.21　消除奇异点后的波形

第 11 章　小波变换在图像处理中的应用

近年来小波分析理论受到众多学科的共同关注。小波变换是传统傅立叶变换的继承和发展。由于小波的多分辨率分析具有良好的空间域和频率域局部化特性，对高频采用逐渐精细的时域或空域步长，可以聚焦到分析对象的任意细节，因此特别适合于图像信号这一类非平稳信源的处理，已成为一种信号/图像处理的新手段。目前，小波分析已被成功地应用于信号处理、图像处理、语音与图像编码、语音识别与合成、多尺度边缘提取和重建、分形及数字电视等科学领域。

本章主要内容包括：

- Matlab 的基本图像类型及相互转换；
- 小波分析在图像消噪中的应用；
- 小波分析在图像压缩中的应用；
- 小波分析在图像平滑中的应用；
- 小波分析在图像增强中的应用；
- 小波分析在图像融合中的应用；
- 小波分析在像字水印中的应用。

11.1　基 本 原 理

11.1.1　基于小波变换的图像分解和重构

利用小波变换的多分辨率特性，可以对图像进行分解和重构。Mallat 基于多分辨率理论，提出了有名的小波变换快速算法 Mallat 算法。为了表达信息便于理解多分辨率的思想，下面给出一个图像分解和重构的例子。基于小波变换的图像三层分解如图 11.1 所示。

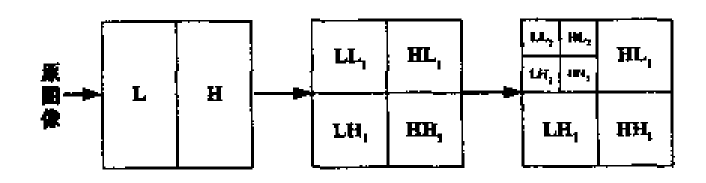


图 11.1　基于小波变换的图像三层分解

在图 11.1 中，原图像首先分解为低频信息 L 和高频信息 H，然后分解低频信息 L 为低频部分 LL_1 和其高频部分 LH_1；对于高频信息 H，重复同样的手段得到 HL_1 和 HH_1。如果需要这个过程可以反复以此类推下去。多分辨率分析的一个最大特点是只对低频空间进行进一步分解，从而使频率的分辨率变得越来越高。二维离散小波分解与重构的示意图分别如图 11.2、

11.3 所示，其中上箭头表示上采样，下箭头表示下采样。

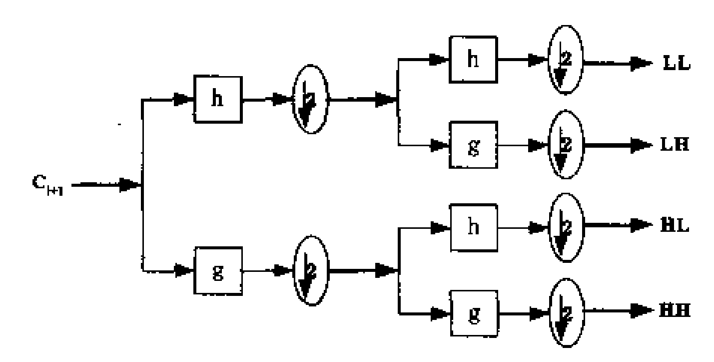


图 11.2　二维离散小波分解示意图

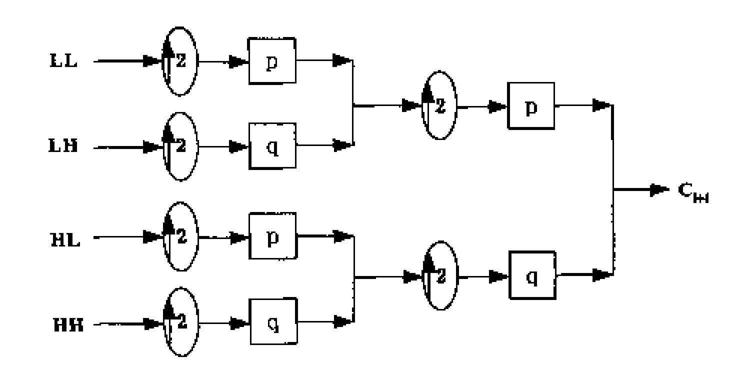


图 11.3　二维离散小波重构示意图

这里我们简要地对上采样的概念进行说明，它是指通过在样本之间插入零值来增加信号分量的过程，如图 11.4 所示。

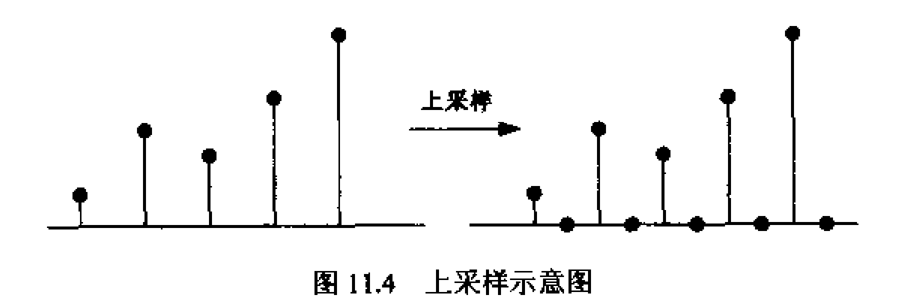


图 11.4　上采样示意图

11.1.2　图像边界的基本处理方法

为了处理信号的边界失真问题，应当有区别地对待信号的边界和其他的部分。通常，在边界上使用基于信号延拓的简单方法是行之有效的，它需要在分解过程中的每一级计算一些额外的系数，因此注意分解过程中的每一级延拓是必需的。下面介绍几种延拓信号的方法。

- 补零

它是指对原始支撑以外的信号以零补足，其缺点是人为地在边界处制造了不连续性。

- 边界对称化

它是指通过对称性的边界值复制来恢复信号（或图像）的原始支撑以外的信号（或图像），这种方法也是小波工具箱中小波变换的缺省模式。其缺点是在边界处，人为地制造了一阶导

数的不连续性，但是这种方法通常对图像处理非常有效。

● 一阶平滑填补

它是指通过简单的一阶导数外插（填补时，对前两个值和后两个值使用线性拟合）来从信号（或图像）的原始支撑之外恢复信号（或图像）。该方法通常对光滑信号较为有效。

● 零阶平滑填补

它是指通过简单的常数外插来从信号（或图像）的原始支撑之外恢复信号（或图像）。对于信号延拓来说，该方法是位于左边的第一个值和右边的最后一个值的重复。

● 第一类周期性填补

它是指通过周期性延拓来恢复信号（或图像）原始支撑以外的信号（或图像），其缺点是在边界处人为地制造了不连续性。

上述 5 种延拓模式均存在一定的冗余，因此对任意一种延拓模式，在其逆变换时，都能确保对信号和图像的完全重构。

● 第二类周期性填补

如果信号的长度是奇数，首先给信号加一个采样点，其值等于最后一个值，接下来，对信号进行第一类周期性填补，即在两端对信号进行最小周期延拓。对于图像，可以采取同样的方法。

这种模式可生成最小长度的小波分解，但是为了确保完全重构，在逆变换中也应采用同样的延拓模式。

11.2 Matlab 基本图像类型及相互转换

11.2.1 Matlab 的基本图像类型

Matlab 的图像类型主要包括索引图像、RGB 图像、二进制图像和灰度图像。

● 索引图像

索引图像包括图像矩阵与颜色图数组，其中颜色图是按图像中颜色值进行排序后的数组。对于每个像素，图像矩阵包含一个值，这个值就是颜色图数组中的索引。颜色图为 m×3 的双精度值矩阵，各行分别对应红、绿、蓝（R、G、B）的单色值，且 R、G、B 均为值域[0，1]上的实数值。

图像矩阵与颜色图的关系依赖于图像矩阵是双精度类型还是无符号 8 位整数类型。如果图像矩阵为双精度类型，则第 1 点的值对应于颜色图的第 1 行，第 2 点的值对应于颜色图的第二行，依此类推，各个点的值都对应于相应颜色图的各个行。如果图像矩阵是无符号 8 位整数类型，且有一个偏移量，即第 0 点的值对应于颜色图的第 1 行，第 1 点的值对应于颜色图的第 2 行，依此类推。无符号 8 位整数类型常用于图形文件格式，支持 256 色。

● RGB 图像

与索引图像一样，RGB 图像也是分别用红、绿、蓝三个亮度值为一组，代表每个像素的颜色。与索引图像不同的是，这些亮度值直接存在图像数组中，而不是存放在颜色图中。图像数组为 m×n×3，m，n 分别表示图像像素的行和列数。

● 二进制图像

在二进制图像中，每个点为两个离散值中的一个，这两个值代表开或关。二进制图像被

保存在一个二维的由 0（关）和 1（开）组成的矩阵中。从另一个角度讲，二进制图像可以看成为一个仅包括黑与白的特殊灰度图像，也可看作仅有两种颜色的索引图像。

二进制图像可以保存为双精度或 unit8 类型的数组，且使用 unit8 类型更节省空间。在图像处理工具箱中，任何一个返回二进制图像的函数都是以 unit8 类型逻辑数组来返回的。

- 灰度图像

在 Matlab 中，灰度图像保存在一个矩阵中，且矩阵中的每一个元素代表一个像素点。矩阵可以是双精度类型，其值域为[0，1]；矩阵也可以是 unit8 类型，其数据范围为[0，255]。矩阵的每一个元素代表不同的亮度或灰度级，其中亮度为 0，表示黑色；亮度为 1（或者 unit8 类型的 255），则代表白色。

11.2.2 Matlab 图像类型的相互转换

本小节对 Matlab 中有关图像类型转换的函数作一简要介绍。

一、灰度图像与索引图像的相互转换

小波分析工具箱中的函数为：gray2ind.m 和 ind2gray.m。

1. gray2ind 函数

【语法格式】

① [X,map] = gray2ind(I,n)

② [X,map] = gray2ind(BW,n)

【使用说明】将灰度图像转化为索引图像。

输入参数：I 是分析的图像；n 是 1～65 536 之间的整数，缺省值为 64；BW 是分析的二进制图像。

输出参数：X 返回转化后的二进制图像；map 返回图像的颜色图 gray(n)。

2. ind2gray 函数

【语法格式】

I=ind2gray(X，map)

【使用说明】将索引图像转化为灰度图像。

输入参数：X 是分析的索引图像，可以是 unit8 或双精度类型；map 是索引图像的颜色图。

输出参数：I 返回索引图像。

二、RGB 图像转换为灰度图像函数

小波分析工具箱中的函数为：rgb2gray.m。

rgb2gray 函数

【语法格式】

① I = rgb2gray(RGB)

② newmap = rgb2gray(map)

【使用说明】将一个 RGB 图像或颜色图转换为灰度图像。

输入参数：RGB 是分析的真彩图像；map 是分析的颜色图。

输出参数：I 和 newmap 返回灰度图。

三、RGB 图像与索引图像的相互转换

小波分析工具箱中的函数为：rgb2ind.m 和 ind2rgb.m。

1. rgb2ind 函数

【语法格式】

① [X,map] = rgb2ind(RGB,tol)

② [X,map] = rgb2ind(RGB,n)

③ X = rgb2ind(RGB,map)

④ [...] = rgb2ind(...,dither_option)

【使用说明】将 RGB 图像转换为索引图像（3 种方法可选）。

输入参数：RGB 是分析的图像，tol 是位于 0～1 之间的数；n 是 1 到 65 536 之间的整数；map 是索引图像的颜色图；dither_option 是 dithering 开关。

输出参数：X 返回索引图像。

2. ind2rgb 函数

【语法格式】

RGB=ind2rgb(X，map)

【使用说明】将矩阵 X（unit8 或双精度格式）及其相应的颜色图 map 转换成 RGB 图像。

输入参数：X 是输入的矩阵；map 是矩阵对应的颜色图。

输出参数：RGB 返回 RGB 图像。

四、其他图像转换为二进制图像

小波分析工具箱中的函数为：im2bw.m。

im2bw 函数

【语法格式】

① BW = im2bw(I,level)

② BW = im2bw(X,map,level)

③ BW = im2bw(RGB,level)

【使用说明】利用阈值将一个图像转换为二进制图像。

输入参数：I 是图像；X 是索引图像；RGB 是 RGB 图像；level 是阈值范围（[0，1]）。

输出参数：BW 返回二进制图像。

使用该函数可将索引、灰度和 RGB 图像转换成为二进制（黑白）图像。在转换过程中，如果输入图像不是灰度级图像，首先将其转换为灰度级图像，然后，通过阈值化将灰度级图像转换成二进制图像。输出图像在输入图像所有亮度小于给定值（level）像素点处均为 0，

在其他地方均为 1。

五、从灰度图像产生索引图像函数

小波分析工具箱中的函数为：grayslice.m。

grayslice 函数

【语法格式】
① X = grayslice(I,n)
② X = grayslice(I,v)
【使用说明】利用多层阈值产生一个索引图像。
输入参数：n 构成阈值 1/n,2/n,…,(n-1)/n；v 为一个元素值在[0，1]上的向量，构成阈值。
输出参数：X 返回索引图像。

11.2.3　例程分析

例程 11-1：不同类型图像之间的转换。

f11_1.m

```
%原始索引图像
load trees

%转换为灰度图像
I = ind2gray(X,map);
figure(1)
subplot(2,2,1);
imshow(X,map)
title('索引图像');
subplot(2,2,2);
imshow(I)
title('转换后的灰度图像');

%原始真彩图像
RGB = imread('peppers.png');

%转换为索引图像
[XX,map] = rgb2ind(RGB,128);
figure(2)
subplot(2,2,1);
imshow(RGB);
title('RGB 图像');
subplot(2,2,2);
imshow(XX,map);
title('转换后的索引图像');
```

```
%原始索引图像
load trees

%转换为二进制图像
BW = im2bw(X,map,0.4);
figure(3)
subplot(2,2,1);
imshow(X,map);
title('索引图像');
subplot(2,2,2);
imshow(BW);
title('转换后的二进制图像');
```

索引图像与灰度图像之间的转换结果如图 11.5 所示。





图 11.5　索引图像与灰度图像之间的转换

RGB 图像与索引图像之间的转换结果如图 11.6 所示。

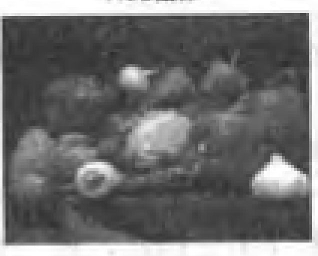




图 11.6　RGB 图像与索引图像之间的转换

索引图像与二进制图像之间的转换结果如图 11.7 所示。





图 11.7　索引图像与二进制图像之间的转换

11.3 小波分析在图像消噪中的应用

图像消噪在信号处理中是一个经典的问题，传统的消噪方法多采用平均或线性方法进行，常用的是维纳滤波，但是消噪效果不够好。随着小波理论的日益完善，它以自身良好的时频特性在图像消噪领域受到越来越多的关注，开辟了用非线性方法消噪的先河。具体说来，小波能够消噪主要得益于小波变换具有如下特点：

● 低熵性：小波系数的稀疏分布，使图像变换后的熵降低。

● 多分辨率特性。由于采用了多分辨率的方法，所以可以非常好地刻画信号的非平稳性，如突变和断点等，可以在不同分辨率下根据信号和噪声的分布来去除噪声。

● 去相关性：小波变换可对信号去相关，且噪声在变换后有白化趋势，所以小波阈比时域更利于去噪。

● 基函数选择灵活：小波变换可以灵活选择基函数，也可根据信号特点和消噪要求选择多带小波、小波包等，对不同的场合，可以选择不同的小波母函数。

11.3.1 基本原理

常用的图像消噪方法是小波阈值消噪方法，它是一种实现简单而效果较好的消噪方法。阈值消噪方法的思想很简单，就是对小波分解后的各层系数模大于和小于某阈值的系数分别进行处理，然后利用处理后的小波系数重构出消噪后的图像。在阈值消噪中，阈值函数体现了对小波分解系数的不同处理策略以及不同估计方法，常用的阈值函数有硬阈值函数和软阈值函数。硬阈值函数可以很好地保留图像边缘等局部特征，但图像会出现伪吉布斯效应等视觉失真现象；而软阈值处理相对较平滑，但可能会造成边缘模糊等失真现象，为此人们又提出了半软阈值函数。

小波阈值消噪方法处理阈值的选取，另一个关键因素是阈值的具体估计。如果阈值太小，消噪后的图像仍然存在噪声；相反如果阈值太大，重要图像特征又将波滤掉，引起偏差。从直观上讲，对给定的小波系数，噪声越大，阈值就越大。

图像信号的小波消噪步骤有三步，和一维信号的消噪步骤完全相同，而所不同的是，处理工具时用二维小波分析代替了一维小波分析工具。因此，对二维图像信号的消噪方法也同样地适合于一维信号，尤其对几何图形更为适合。二维小波分析用于图像消噪的步骤如下。

步骤 1：二维图像信号的小波分解。

在这一步，应当选择合适的小波和恰当的分解层次（记为 N），然后对待分析的二维图像信号 X 进行 N 层分解计算。

步骤 2：对分解后的高频系数进行阈值量化。

对于分解的每一层，选择一个恰当的阈值，并对该层高频系数进行软阈值量化处理。在此，阈值选取规则同前面的信号处理部分。

步骤 3：二维小波的重构图像信号。

同样地，根据小波分解后的第 N 层近似（低频系数）和经过阈值量化处理后的各层细节

（高频系数），来计算二维信号的小波重构。

下面通过具体的例子来说明利用小波分析进行图像消噪这个问题。

11.3.2　例程分析

例程 11-2：利用小波分析对给定一个二维含噪图像进行消噪处理。

f11_2.m

```
%装载原始图像信号
load wgatlin;
subplot(2,2,1);
image(X);
colormap(map);
title('原始图像');

%生成含噪图像
init=2055615866;
rand('seed',init);
XX=X+12*randn(size(X));
subplot(2,2,2);
image(XX);
colormap(map);
title('含噪图像');

%下面对图像消噪
%首先用 sym5 小波函数对图像进行 2 层分解
[c,l]=wavedec2(XX,2,'sym5');
%图像第一层的重构逼近信号
a1=wrcoef2('a',c,l,'sym5',1);
%图像第二层的重构逼近信号
a2=wrcoef2('a',c,l,'sym5',2);

%图示消噪处理后的结果
subplot(2,2,3);
image(a1);
colormap(map);
title('第一层重构图像');

subplot(2,2,4);
image(a2);
colormap(map);
title('第二层重构图像');
```

利用小波变换对含噪的原始图像进行消噪，结果如图 11.8 所示。

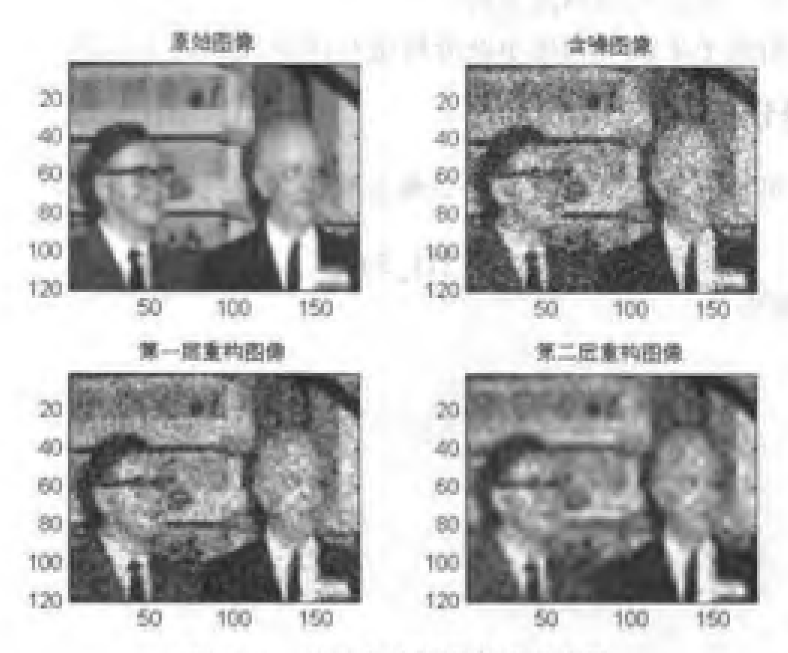


图 11.8 基于小波的图像消噪结果

例程 11-3：利用二维小波变换对给定图像进行小波消噪处理。

f11_3.m

```
%装载并图示原始图像
load gatlin2;
subplot(2,2,1);
image(X);
colormap(map);
title('原始图像');

%生成含噪图像并图示
init=2055615866;
randn('seed',init);
XX=X+8*randn(size(X));
subplot(2,2,2);
image(XX);
colormap(map);
title('含噪图像');

%对图像进行消噪处理
%用小波函数 coif2 对图像 XX 进行 2 层分解
[c,l]=wavedec2(XX,2,'coif2');
%设置尺度向量
n=[1,2];
%设置阈值向量
p=[10.28,24.08];
```

```
%对高频小波系数进行阈值处理
%nc=wthcoef2('h',c,l,n,p,'s');
%nc=wthcoef2('v',c,l,n,p,'s');
nc=wthcoef2('d',c,l,n,p,'s');
%图像的二维小波重构
X1=waverec2(nc,l,'coif2');
subplot(2,2,3);
image(X1);
colormap(map);
title('第一次消噪后的图像');

%再次对高频小波系数进行阈值处理
%mc=wthcoef2('h',nc,l,n,p,'s');
mc=wthcoef2('v',nc,l,n,p,'s');
%mc=wthcoef2('d',nc,l,n,p,'s');

%%图像的二维小波重构
X2=waverec2(mc,l,'coif2');
subplot(2,2,4);
image(X2);
colormap(map);
title('第二次消噪后的图像');
```

利用不同阈值对含噪图像消噪的结果如图 11.9 所示。第一次消噪滤去了大部分高频噪声，但与原图比较，依然有不少高频噪声。第二次消噪在第一次消噪基础上，再次滤去高频噪声，消噪效果较好，但图像质量比原图稍差。

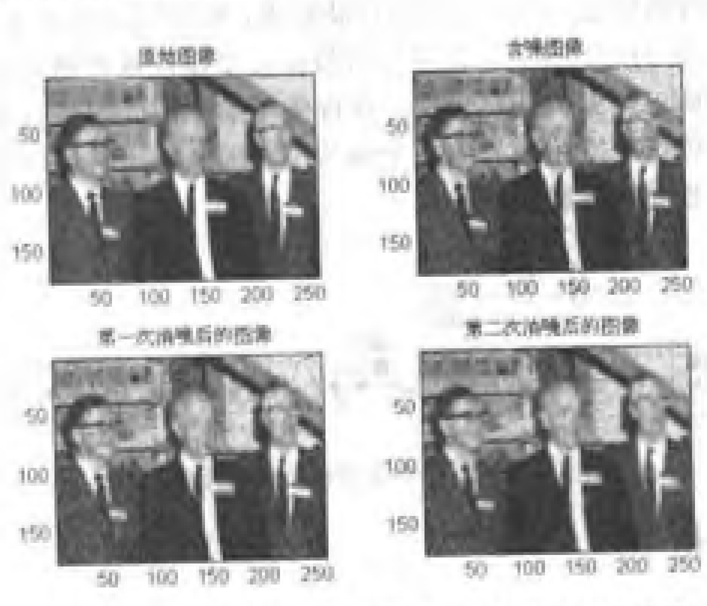


图 11.9　基于小波（设置不同阈值）的图像消噪结果

11.4　小波分析在图像压缩中的应用

在分布式网络多媒体应用中，为了达到令人满意的画面和听觉效果，需要对视频信号和音频信号进行实时处理，因此对数据实现高保真、大压缩比的压缩非常必要。数据压缩主要包含无失真压缩和有失真压缩两大类。无失真压缩是指图像数据经压缩后可以完全地得到恢复，复原后的图像和原始图像一致。而有失真压缩是指经过压缩后的图像数据在保持原图像特征的前提下，不可避免地要丢失一部分不重要的图像原始信息。目前基于小波变换的图像压缩方法已经逐步取代基于离散余弦或者其他子代编码技术，而成为新的图像压缩国际标准的首选方法。

11.4.1　基本原理

图像数据之所以能够进行压缩，其数学机理主要有下面两点：

（1）原始图像数据往往存在各种信息的冗余（如空间冗余、视觉冗余和结构冗余等），数据之间存在相关性，邻近像素的灰度（将其看成随机变量）往往是高度相关的。

（2）在多媒体应用领域中，人眼作为图像信息的接受端，其视觉对于边缘急剧变化不敏感，以及人眼存在对图像的亮度信息敏感，而对颜色分辨率弱等，因此在高压缩比的情况下，解压缩后的图像信号仍有满意的主观质量。

所谓图像压缩就是去掉各种冗余，保留重要的信息。图像压缩的过程常称为编码，而图像的恢复则称为解码。

虽然图像的数据是非常巨大的，但是可以采用适当的坐标变换去除相关，从而达到压缩数据的目的。传统的 K.L 变换就是以这种思想为基础的，它把信号的一小块看成是一个独立的随机向量，它的基函数由余弦函数组成。

小波变换通过多分辨分析过程将一幅图像分成近似和细节两部分，细节对应的是小尺度的瞬变，它在本尺度内很稳定。因此将细节存储起来，对近似部分在下一个尺度上进行分解，重复该过程即可，如图 11.10 所示。近似与细节在正交镜像滤波器算法中分别对应于高通和低通滤波，这种变换通过尺度去掉相关性，在视频压缩中被证明是有效的。

上述方法有一个优点：即图像分成多个细节层，因此应用时可以先给出一幅较为粗糙的图像，然后可根据需要提供更好的细节。

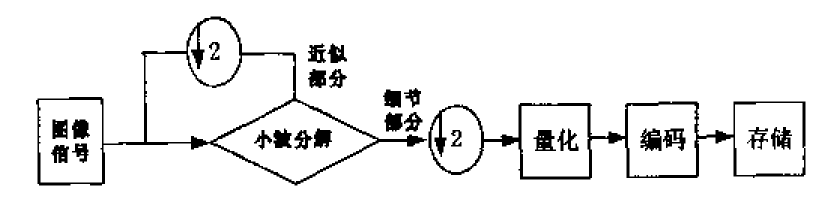


图 11.10　基于小波变换的图像压缩过程

11.4.2　例程分析

对图像作小波分解后，可得到一系列不同分辨率的子图像（它们所对应的频率不相同）。

而对于图像来说，表征它的最主要部分是低频部分，而高频部分大部分点的数值均接近于 0，而且频率越高，这种现象越明显。因此，利用小波分解去掉图像的高频部分而仅保留图像的低频部分是一种最简单的图像压缩方法。

例程 11-4：利用二维小波分析对给定图像进行压缩。

f11_4.m

```
%装载并显示原始图像
load woman;
subplot(2,2,1);
image(X);
colormap(map);
title('原始图像 ');
axis square;
disp('压缩前图像的大小：');
whos('X')

%对图像进行 7 层小波分解
[c,l]=wavedec2(X,2,'bior3.7');
%提取小波分解结构中的一层的低频系数和高频系数
cA1=appcoef2(c,l,'bior3.7',1);
%水平方向
cH1=detcoef2('h',c,l,1);
%斜线方向
cD1=detcoef2('d',c,l,1);
%垂直方向
cV1=detcoef2('v',c,l,1);

%重构第一层系数
A1=wrcoef2('a',c,l,'bior3.7',1);
H1=wrcoef2('h',c,l,'bior3.7',1);
D1=wrcoef2('d',c,l,'bior3.7',1);
V1=wrcoef2('v',c,l,'bior3.7',1);
c1=[A1 H1;V1 D1];

%显示第一层频率信息
subplot(2,2,2);
image(c1);
title('分解后的低频和高频信息');

%对图像进行压缩：保留第一层低频信息并对其进行量化编码
ca1=wcodemat(cA1,440,'mat',0);
%改变图像高度并显示
ca1=0.5*ca1;
```

```
subplot(2,2,3);
image(ca1);
colormap(map);
title('第一次压缩后图像');
axis square;
disp('第一次压缩后图像的大小：');
whos('ca1')
%压缩图像：保留第二层低频信息并对其进行量化编码
cA2=appcoef2(c,l,'bior3.7',2);
ca2=wcodemat(cA2,440,'mat',0);
ca2=0.5*ca2;
subplot(2,2,4);
image(ca2);
colormap(map);
title('第二次压缩后图像');
disp('第二次压缩后图像大小：');
whos('ca2')
```

压缩后的图像如图 11.11 所示。

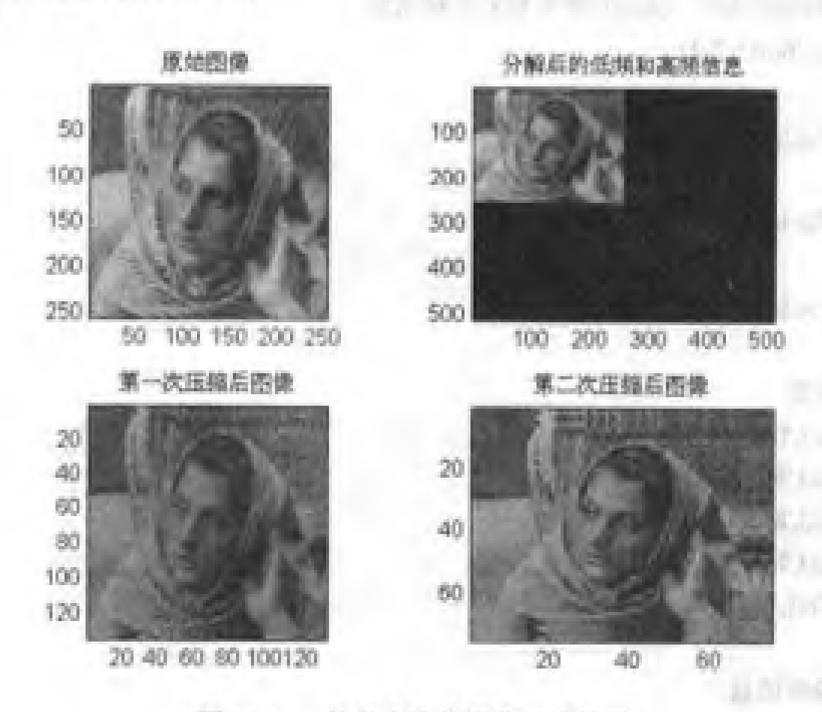


图 11.11　基于小波的图像压缩结果

压缩前图像的大小：

```
Name        Size          Bytes    Class
X         256×256         524288   double array
Grand total is 65536 elements using 524288 bytes
```

第一次压缩后图像的大小：

```
Name        Size          Bytes    Class
```

```
ca1        135×135        145800    double array
Grand total is 18225 elements using 145800 bytes
```

第二次压缩后图像大小：

```
Name        Size          Bytes      Class
ca2         75×75         45000      double array
Grand total is 5625 elements using 45000 bytes
```

在这里可以看出，第一次压缩我们是提取原始图像中小波分解第一层的低频信息，此时压缩效果较好，压缩比较小（约为 1/3 大小）。第二次压缩是提取第一层分解低频部分的低频部分（即第二层的低频部分），其压缩比较大（约为 1/12），压缩效果在视觉上也基本过得去。随着分解层数的增加，压缩比是递减的。

上面的保留原始图像中低频信息的压缩办法只是一种最简单的压缩办法。它不需经过其他处理即可获得较好的压缩效果。当然，对于上面的例子我们还可以只提取小波分解的第三、第四层的低频信息。从理论上说，我们可以获得任意压缩比的压缩图像。只不过在对压缩比和图像质量都有较高要求时，它就不如其他编码方法了。

在 Matlab 的小波工具箱中，有专门利用阈值压缩图像的函数 wdencmp。下面我们就通过一个具体的例子来说明该函数的应用及其特点。

例程 11-5：利用函数 wdencmp 对给定图像进行压缩处理。

f11_5.m

```
%装载并显示原始图像
load flujet;
subplot(1,2,1);
image(X);
colormap(map);
title('原始图像');

%首先利用 db3 小波对图像 X 进行 2 层分解
[c,l]=wavedec2(X,2,'db3');

%全局阈值
[thr,sorh,keepapp]=ddencmp('cmp','wv',X);

%压缩处理：对所有高频系数进行同样的阈值量化处理
[Xcmp,cxc,lxc,perf0,perfl2]=wdencmp('gbl',c,l,'db3',2,thr,sorh,keepapp);

%将压缩后的图像于原始图像相比较
subplot(1,2,2);
image(Xcmp);colormap(map);
title('压缩后的图像');

%显示相关参数
disp('小波分解系数中为 0 的系数个数百分比：');
```

```
perf0
disp('压缩后保留能量百分比：');
perfl2
```

结果如下：首先给出应用函数 wdencmp 进行压缩的效率，即分解系数中置 0 的系数个数百分比和保留能量百分比。压缩图像如图 11.12 所示。

小波分解系数中为 0 的系数个数百分比：

perf0 =

89.5829

压缩后保留能量百分比：

perfl2 =

99.9831

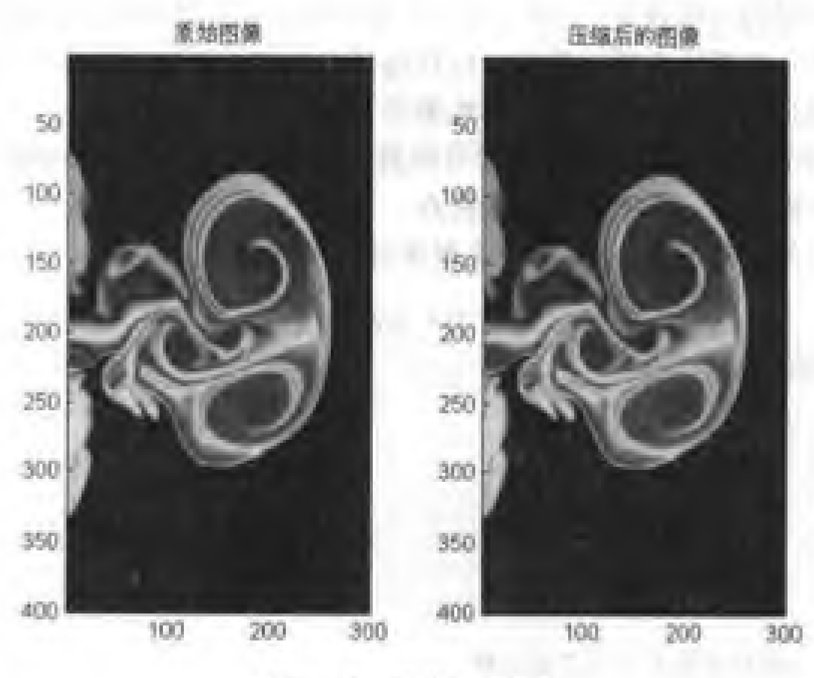


图 11.12 图像压缩结果

11.5 小波分析在图像平滑中的应用

11.5.1 基本原理

图像平滑的主要目的是为了减少噪声。一般情况下，在空间域内可以用平均来减少噪声。而在频率域，由于噪声多在高频段，因此可以使用各种形式的低通滤波办法来减少噪声。

11.5.2 例程分析

例程 11-6：利用二维小波分析和图像的中值滤波对一给定的含噪图像进行平滑处理。

f11_6.m

```
%装载原始图像
```

```
load gatlin;

%对图像加噪声并显示出含噪图像
init=2788605800;
randn('seed',init);
X=X+10*randn(size(X));
subplot(2,2,1);
image(X);
colormap(map);
title('含噪图像');

%应用中值滤波进行图像平滑处理
[p,q]=size(X);
for i=2:p-1
    for j=2:q-1
        Xtemp=0;
        for m=1:3
            for n=1:3
                Xtemp=Xtemp+X(i+m-2,j+n-2);
            end
        end
        Xtemp=Xtemp/9;
        X1(i,j)=Xtemp;
    end
end

%显示结果
subplot(2,2,2);
image(X1);
colormap(map);
title('平滑后图像');
```

利用小波变换对图像的平滑结果如图 11.13 所示。

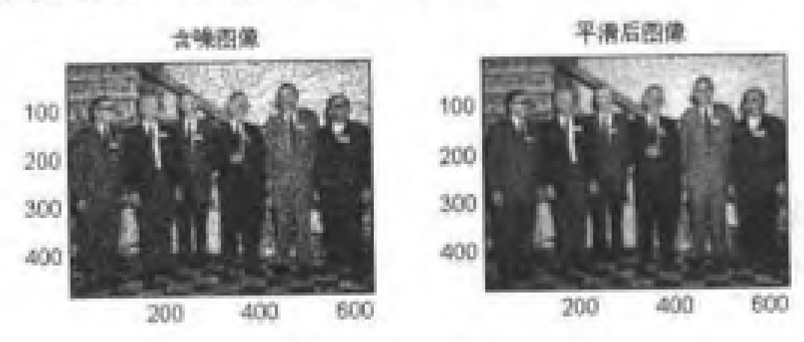


图 11.13　基于小波的图像平滑结果

在本例程中，我们首先对图像加了一个较大的白噪声，之后应用中值滤波对含噪图像进

行处理。从图 11.13 中可以看出，含噪图像经过基于小波的中值滤波处理后具有较好的平滑效果。

11.6 小波分析在图像增强中的应用

数字图像是通过对实际连续的图像按照一定的间隔采样得到的离散化的数据，一般说来，摄影、扫描等成像设备得到的图像质量都不是很高，其中一个重要的原因是可用灰度级没有得到有效使用，图像对比度不够，视觉上比较模糊。为了便于人眼和机器识别，要求提高图像的对比度，图像增强就是针对这一问题的处理技术。

需要说明的是，增强技术不能增加图像数据本身所含的信息，但是可以凸现特定的特征，处理更有利于识别，从而有利于特定的应用。通常意义上的图像增强目标主要是放大图像中感兴趣结构的对比度，增加可理解性，同时减少或抑制图像中混有的噪声，提高视觉质量。

11.6.1 基本原理

按照处理空间的不同，常用的增强技术可以分为基于图像域和机遇变幻域两种。前一种方法直接对像素点进行运算，而后一种方法相对比较复杂，它首先将图像从空间域变换到另一个域内表示（最常用的是时域到频域的变换），通过修正相应域内系数达到提高输出图像对比度的目的。

增强是图像处理中最基本的技术之一，这里只介绍基于多尺度方法的增强技术。小波变换将一幅图像分解为大小、位置和方向均不相同的分量，在作逆变换之前，可根据需要对不同位置、不同方向上的某些分量改变其系数的大小，从而使得某些感兴趣的分量放大而使某些不需要的分量减小。

下面我们用具体的例子来说明图像的增强。

11.6.2 例程分析

例程 11-7：用小波分析对所给图像进行增强处理。

f11_7.m

```
%装载并显示原始图像
load flujet;
subplot(1,2,1);
image(X);
colormap(map);
title('原始图像');

%对图像 X 用小波 db5 进行 2 层分解
[c,l]=wavedec2(X,2,'db5');
Csize=size(c);

%弱化不重要的分解系数
for i=1:Csize(2)
```

```
    if(c(i)>300)
        c(i)=2*c(i);
    else
        c(i)=0.5*c(i);
    end
end

%重构图像并显示
X1=waverec2(c,l,'db5');
subplot(1,2,2);
image(X1);
colormap(map);
title('增强后图像');
```

结果如图 11.14 所示。

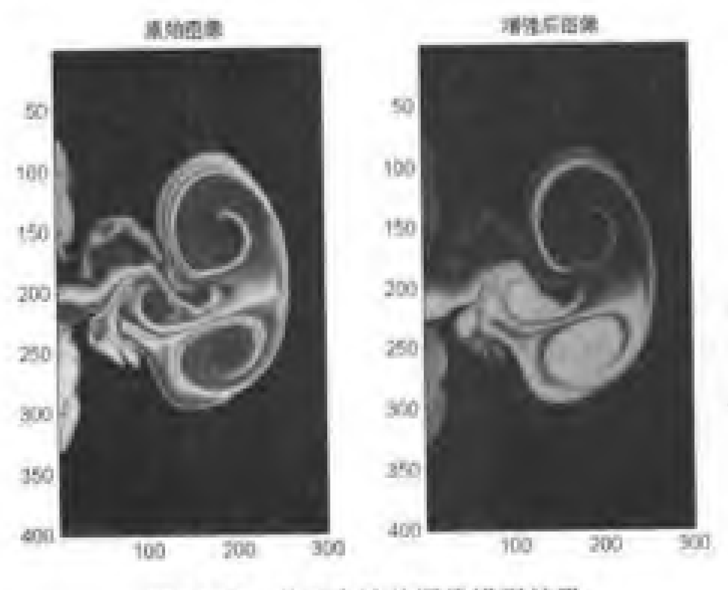


图 11.14　基于小波的图像增强结果

分解后的图像，其主要信息（即轮廓）由低频部分来表征，而其细节部分则由高频部分表征。因此，在上述的例子中，我们对分解后的低频系数加权进行增强，而对高频部分加权进行弱化，经过如此处理后，即达到了增强图像的目的。

11.7　小波分析在图像融合中的应用

11.7.1　基本原理

图像融合是将同一对象的两个或更多的图像合成在一幅图像中，以便它比原来的任何一幅更能容易的为人们所理解。这一技术可应用于多频谱图像理解以及医学图像处理等领域，在这些场合，同一物体部件的图像往往是采用不同的成像机理得到的。

11.7.2 例程分析

例程 11-8：用小波分析对两个不同的图像进行融合。

f11_8.m

```
%装入原图像 1
load woman;
X1=X;
map1=map;
subplot(221);
%画出 woman 图像
image(X1);
colormap(map1);
title('原图像 1');
%axis square

%装入原图像 2
load wbarb;
X2=X;
map2=map;
for I =1:256
    for j=1:256
        if(X2(I, j)>100)
            X2(I, j)=1.2*X2(I, j);
        else
            X2(I, j)=0.5*X2(I, j);
        end
    end
end

subplot(222);
%画出 wbarb 图像
image(X2);
colormap(map2);
title('原图像 2');
%axis square

[c1,s1]=wavedec2(X1,2,'sym4');
sizec1=size(c1);
for I=1:sizec1(2)
    c1( I )=1.2*c1( I );
end

[c2,s2]=wavedec2(X2,2,'sym4');
```

```
c=c1+c2;
c=0.5*c;
s=s1+s2;
s=0.5*s;
xx=waverec2(c,s,'sym4');

subplot(223);
%画出融合后的图像
image(xx);
title('融合图像');
%axis square
```

两个原始图像及其融合的结果如图 11.15 所示。

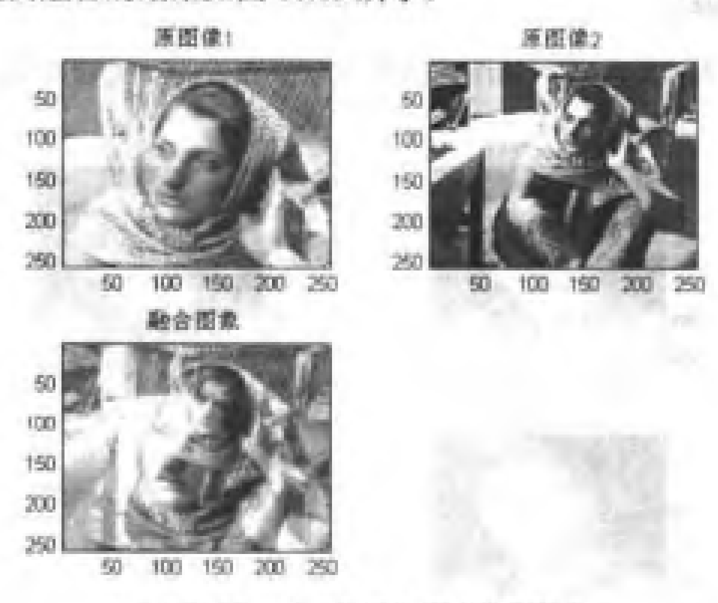


图 11.15　基于小波的图像融合结果

一幅图像和它某一部分放大后的图像融合，融合后的图像给人一种朦朦胧胧梦幻般的感觉，对较深的背景部分则做了淡化处理。

例程 11-9：利用图像融合方法从模糊图像中恢复图像。

f11_9.m

```
%调入第一幅模糊图像
load cathe_1;
X1=X;
%调入第二幅模糊图像
load cathe_2;
X2=X;
%基于小波分解的图像融合
XFUS=wfusimg(X1,X2,'sym4',5,'max','max');

%显示
```

```
colormap(map);
subplot(2,2,1);
image(X1);
axis square;
title(' Catherine 1');
subplot(2,2,2);
image(X2);
axis square;
title(' Catherine 2');
subplot(2,2,3);
image(XFUS);
axis square;
title('Synthesized image');
```

利用图像融合从两幅模糊的原始图像中恢复出原图像的结果如图 11.16 所示。



图 11.16　利用图像融合恢复图像

11.8　小波变换在数字水印中的应用

数字水印技术是目前信息安全技术领域的一个新方向，是一种可以在开放的网络环境下保护版权和认证来源及完整性的新型技术，创作者的创作信息和个人标志通过数字水印系统以人所不可感知的水印形式嵌入在多媒体中。人们无法从表面上感知水印，只有专用的检测器或计算机软件才可以检测出隐藏的数字水印。

数字水印技术的研究发展目前可分为两代。第一代技术主要研究在保证水印不可感知性的前提下，提高数字水印对多媒体的一般处理、加噪、有损压缩等攻击的鲁棒性；第二代技术在第一代的基础上，重点研究了增强数字水印对仿射变换和几何攻击（RST）等的抵抗能力，并提出了一些新的水印技术和方法。

实现水印技术的典型算法有：

（1）最低有效位算法（LSB）：它是国际上最早提出的数字水印算法，是一种典型的空间域信息隐藏算法。它可以隐藏较多的信息，但当受到各种攻击后水印很容易被移去。

（2）Patchwork 算法：是麻省理工学院媒体实验室提出的一种数字水印算法，主要用于打印票据的防伪。其缺陷是所隐含的数据量较少，对仿射变换敏感。

（3）基于 DCT 的频域水印算法：这是目前研究最多的算法。它具有鲁棒性强、隐蔽性好等特点，尤其可以与 JPEG、MPEG 等压缩标准的核心算法相结合，能较好地抵抗有损压缩。

（4）扩展频谱方法：是扩频通信技术在数字水印中的应用，其特点是应用一般的滤波手段无法消除水印。

（5）小波变换（WT）算法：已有一些学者研究了在小波域中隐藏数字水印信息的算法，并取得了较好的效果。

11.8.1　基本原理

水印的基本原理是嵌入某些标识数据到宿主数据中作为水印，使得水印在宿主数据中不可感知和足够安全。为了保证由于水印的嵌入而导致宿主数据失真不被察觉到，必须应用某种感知准则，不管是隐性还是显性。水印算法要结合加密方法以提供其安全性，通用的水印算法包含两个基本方面：水印的嵌入和水印的提取。水印可以由多种模型构成，如随机数字序列、数字标识、文本以及图像等。

数字水印的嵌入过程如图 11.17 所示：

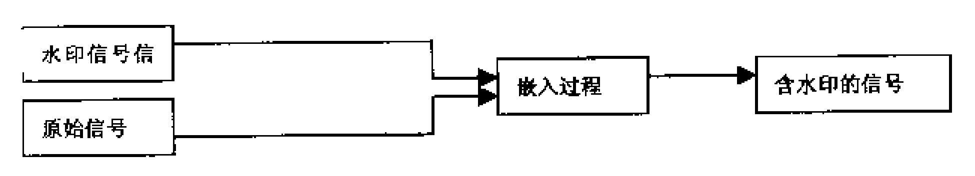


图 11.17　数字水印嵌入过程

频域法加入数字水印的原理是首先将原始信号（语音一维信号、图像二维信号）变换到频域，常用的变换一般有 DWT、DCT、DFT、WP 和分形。然后，对加入了水印信息的信号进行频域反变换（IDWT、IDCT、DFT、WP），得到含有水印信息的信号。

数字水印的检测过程如图 11.18 所示：

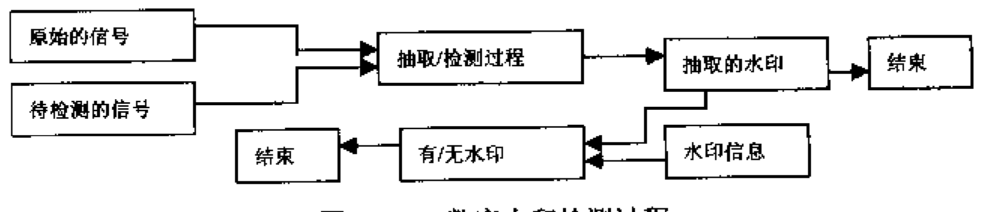


图 11.18　数字水印检测过程

频域法检测水印的原理是将原始信号与待检测信号同时进行变换域变换，比较两者的区别，进行嵌入水印的逆运算，得出水印信息。如果是可读的水印，那么就此结束，如果是不可读水印，如高斯噪声，就将得出的水印与已知水印作比较，由相关性判断，待检测信号含不含水印，故水印的检测有两个结束点。

下面介绍一种典型的基于小波变换的数字水印方法。

（1）第一步，将水印图像作时域上的变换，目的是对水印信息进行乱序，达到加密的效果。采用函数：

$$A_N(k):\begin{bmatrix} x' \\ y' \end{bmatrix}=\begin{bmatrix} 1 & 1 \\ k & k+1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \quad \text{mod N}$$

其中 k 是一个控制参数，N 是矩阵的大小，(x,y) 和 (x,y) 表示像素点在变换前后的位置。假设 P 表示由二值水印信息组成的一个 $m\times m$ 的矩阵，对每一个点的坐标作 $A_N(k)$变换之后，这个 $m\times m$ 的矩阵将变成一个 $N\times N$ 的矩阵，矩阵的每个元素为 0 或 1。

（2）第二步，对图像作小波变换，对于变换后得到的小波系数，选出一个起始位置在 (P_1, P_2)，大小为 $N\times N$ 得系数矩阵。这个矩阵的大小与水印图像作时域变换后形成的矩阵大小是一致的。

（3）第三步，在选出的系数矩阵中嵌入水印信息，即将两个 $N\times N$ 的矩阵进行信息叠加，其中含有水印信息的矩阵元素为 0 或 1。TYC 提出了一种信息叠加的方案。

记：

A——水印信息进行时域变换后得到的大小为 $N\times N$ 的矩阵；

U——在矩阵 A 中含有水印信息的位置的集合；

B——图像经过小波变换后得到的系数矩阵（$N\times N$）；

S——模；

C——B 和 U 的交集：

$$\delta(i,j)——\delta(i,j)=c(i,j)\text{modS};$$

对于所有属于 U 和 A 交集的点 $c(i,j)$：

如果 $A(i,j)=1$，并且 $B(i,j)\geqslant 0$，则 $c(i,j)= c(i,j)-\delta(i,j)+T_1$；

如果 $A(i,j)=0$，并且 $B(i,j)\geqslant 0$，则 $c(i,j)= c(i,j)-\delta(i,j)+T_2$；

如果 $A(i,j)=1$，并且 $B(i,j)<0$，则 $c(i,j)= c(i,j)+\delta(i,j)-T_1$；

如果 $A(i,j)=0$，并且 $B(i,j)<0$，则 $c(i,j)= c(i,j)+\delta(i,j)-T_2$；

这里 T_1,T_2 是水印嵌入的门限，安全性系数包括 $n,k,p_1,p_2,m,N,S,T_1,T_2$。

水印的提取过程如下：

假设 y 是从小波变换域抽取的一个 $N\times N$ 的系数矩阵，起始位置为 (P_1,P_2) ：$\theta(i,j)$ 满足：$\theta(i,j)=Y(i,j)\text{modS}$，D 是一个 $N\times N$ 的矩阵。对 Y 中的所有点 (i,j)，定义

如果$|\theta(i,j)|\geqslant(T_1+T_2)/2$，则 $D(i,j)=1$；

如果$|\theta(i,j)|<(T_1+T_2)/2$，则 $D(i,j)=1$；

因此对矩阵 D 作 T-n 次 $A_N(k)$ 反变换，水印图像就被恢复出来了。

11.8.2 例程分析

例程 11-10：利用二维离散小波变换实现数字水印。

f11_10.m

```
clear;
%装入原图像 1
load woman;
I=X;
```

```
%小波函数
type = 'db1';

% 2 维离散 Daubechies 小波变换
[CA1, CH1, CV1, CD1] = dwt2(I,type);
C1 = [CH1 CV1 CD1];

%系数矩阵大小
[length1, width1] = size(CA1);
[M1, N1] = size(C1);

% 定义阈值 T1
T1 =50;
alpha = 0.2;

%在图像中加入水印
for counter2 = 1: 1: N1
    for counter1 = 1: 1: M1
        if( C1(counter1, counter2) > T1 )
            marked1(counter1,counter2) = randn(1,1);
            NEWC1(counter1, counter2) = double( C1(counter1, counter2) ) +...
alpha * abs( double( C1(counter1, counter2) ) ) * marked1(counter1,counter2) ;
        else
            marked1(counter1, counter2) = 0;
            NEWC1(counter1, counter2) = double( C1(counter1, counter2) );
        end;
    end;
end;

%重构图像
NEWCH1 = NEWC1(1:length1, 1:width1);
NEWCV1 = NEWC1(1:length1, width1+1:2*width1);
NEWCD1 = NEWC1(1:length1, 2*width1+1:3*width1);

R1 = double( idwt2(CA1, NEWCH1, NEWCV1, NEWCD1, type) );

%分离水印
watermark1 = double(R1) - double(I);

figure(1);
subplot(1,2,1);
image(I);
```

```
axis('square');
title('原始图像');
subplot(1,2,2);
imshow(R1/250);
axis('square');
title('Daubechies 小波变换后图像');

figure(2);
imshow(watermark1*10^16);
axis('square');
title('水印图像');

% 水印检测
newmarked1 = reshape(marked1, M1*N1, 1);
% 检测阈值
T2 = 60;
 for counter2 = 1: 1: N1
          for counter1 = 1: 1: M1
            if( NEWC1(counter1, counter2) >T2 )
                NEWC1X(counter1, counter2) = NEWC1(counter1, counter2);
            else
                NEWC1X(counter1, counter2) = 0;
            end;
        end;
    end;

NEWC1X = reshape(NEWC1X, M1*N1, 1);

correlation1 = zeros(1000,1);
for corrcounter = 1: 1: 1000
    if( corrcounter == 500)
      correlation1(corrcounter,1) = NEWC1X'*newmarked1 / (M1*N1);
    else
      rnmark = randn(M1*N1,1);
      correlation1(corrcounter,1) = NEWC1X'*rnmark / (M1*N1);
    end;
end;

% 计算阈值
originalthreshold = 0;
for counter2 = 1: 1: N1
          for counter1 = 1: 1: M1
             if( NEWC1(counter1, counter2) > T2 )
```

```
                originalthreshold = originalthreshold + abs( NEWC1(counter1, counter2) );
            end;
        end;
    end;
originalthreshold = originalthreshold * alpha / (2*M1*N1);

corrcounter = 1000;
originalthresholdvector = ones(corrcounter,1) * originalthreshold;

figure(3);
plot(correlation1, '-');
hold on;
plot(originalthresholdvector, '--');
title('原始的加水印图像');
xlabel('水印');
ylabel('检测响应');
```

原始图像和小波变换后的图像如图 11.19 所示，加入的水印图像如图 11.20 所示，水印检测结果如图 11.21 所示。



图 11.19　原始图像和小波变换后的图像



图 11.20　水印图像

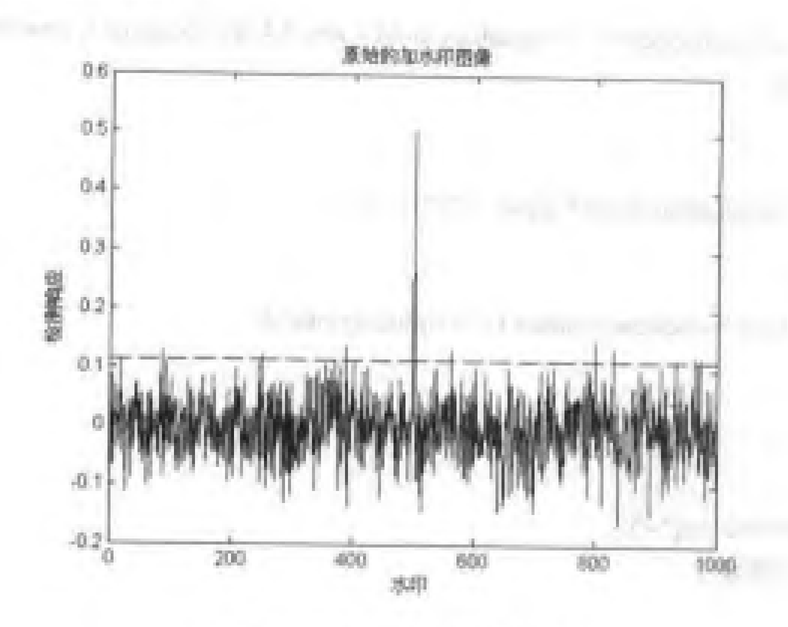


图 11.21　水印图像的检测结果

第 12 章　小波包在信号处理中的应用

小波包分析属于线性时频分析法，它具有良好的时频定位特性以及对信号的自适应能力，因而能够对各种时变信号进行有效的分解。

本章的主要内容包括：

- 小波包树的管理；
- 小波包在信号时频分析中的应用；
- 小波包在信号消噪中的应用；
- 小波包在信号压缩中的应用。

12.1　引　　言

实际上，在许多问题中我们只是对某些特定时间段或频率段的信号感兴趣，只要提取这些特定时间或频率点上的信息即可。因此我们自然希望在感兴趣的频率点上最大可能地提高频率分辨率，在感兴趣的时间点上最大程度地提高时间分辨率，而正交小波变换所提供的按固定规律变化的时频相平面将不再满足这种要求，其主要原因在于正交小波变换的多分辨率分解只将尺度空间进行了分解，而没有对小波空间进行进一步的分解。而通过对小波空间进行进一步的分解，可使正交小波变换的频谱窗口进一步分割变细，这就是小波包变换的基本思想。

小波包方法是对信号进行时频分解的一种方法，它具有对信号特征的自适应性，因而能够有效地显示信号的时频特征，小波包分解是通过正交镜像滤波器进行的。假设信号为 $y(t)$，则有以下递推公式：

$$\begin{cases} y_{2n}(t) = \sqrt{2}\sum_{k} h(k) y_n(2t-k) \\ y_{2n+1}(t) = \sqrt{2}\sum_{k} g(k) y_n(2t-k) \end{cases}$$

函数系 $\{y_n(t)\}$ 称作正交小波包，它是原信号在各种尺度上所有频段的全部分解结果。令 $k=n-2^j$，则 $y_n(t)=y_{2^j+k}(t)$ 为信号对于尺度 j 在频段 k 上的分解结果。

小波包可以组成许多不同正交基的分解结果，其中比较典型的有小波基和子波基，对于所有的组合选取熵最小者，即得到最佳基。最佳基的分解结果最能表征信号的时频特性，因而体现了该方法对于信号的自适应性。

信号经过小波包分解并选出最优基后，需要将最佳基上的分解结果在时频面上表示出来。如果原离散信号的样本数为 N，则分解结果可以表示为时频面上 N 个面积为 $\Delta t \times f\Delta$ 的相邻小矩形（Δt 和 Δf 分别为时间和频率分辨率）。

12.2 小波包树的管理

12.2.1 从命令行显示小波包树

例程 12-1：对信号进行小波包分解后，观察其树结构。

f12_1.m

```
%读信号
load noisbump
x = noisbump;

%3 层小波包分解
t = wpdec(x,3,'db2');

%显示小波包树结构
fig = plot(t);
```

计算得到的小波包树结构如图 12.1 所示。

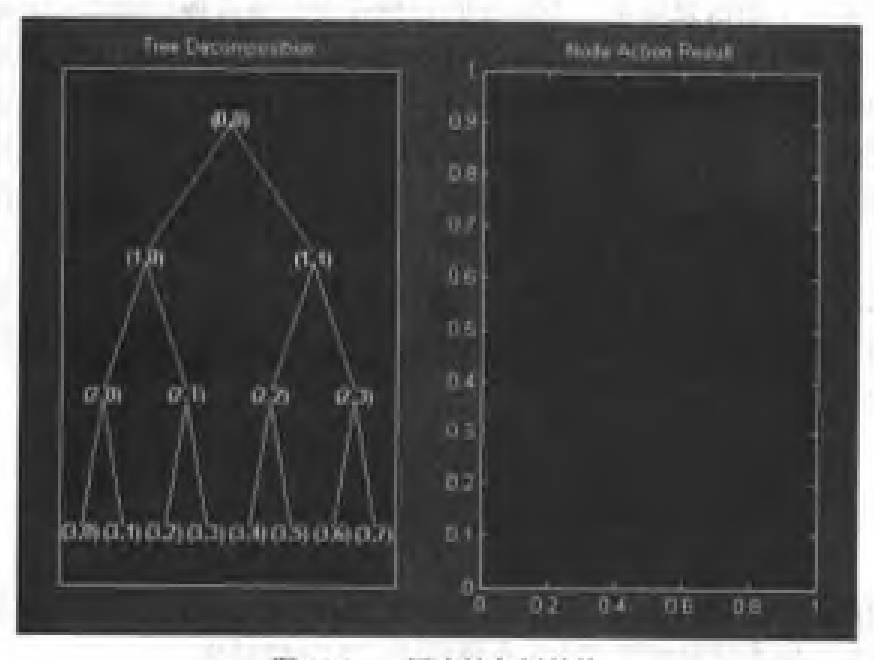


图 12.1　3 层小波包树结构

在菜单“Node Label”下选择“Index”，然后点击左边的节点“(14)”，显示其信号波形，得到的结果如图 12.2 所示。

在菜单“Node Action”下选择“Split/Merge”，然后点击节点“(2)”合并节点 2，接着在菜单“Node Action”下选择“Visualize”，再点击节点“(2)”，显示其信号波形如图 12.3 所示。

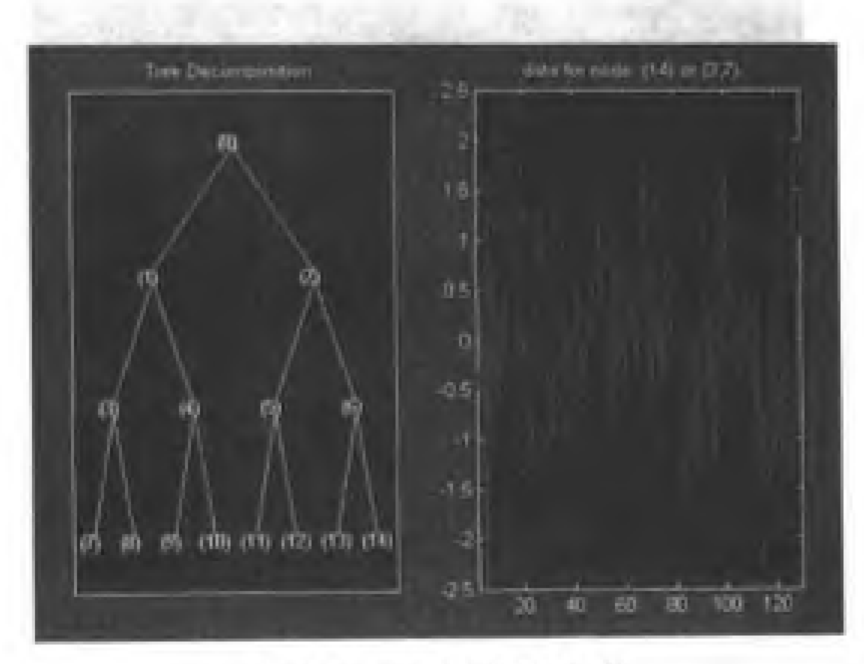


图 12.2　显示节点的信号波形

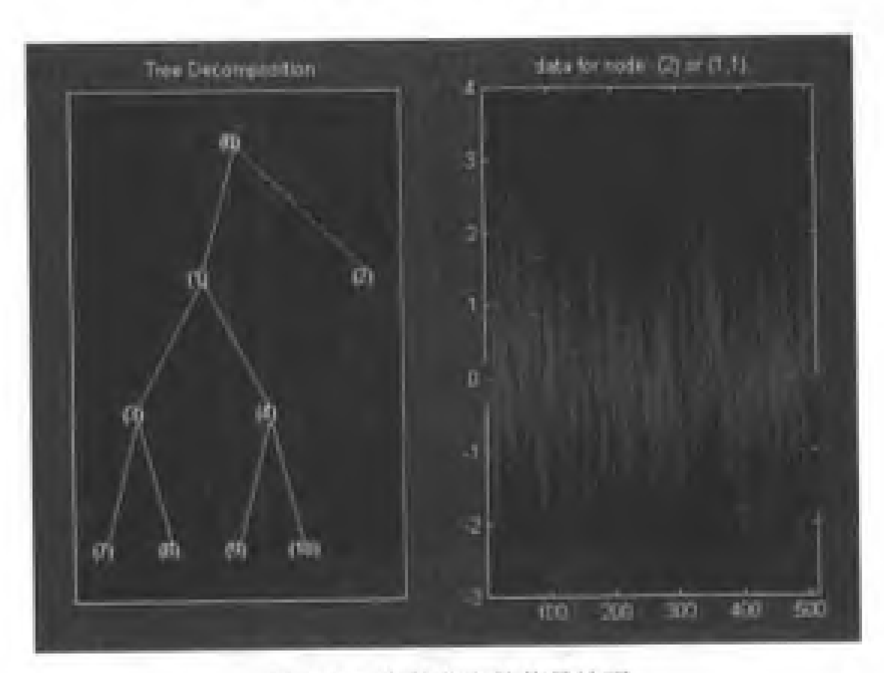


图 12.3　合并节点的信号波形

得到新的树结构后，可以对其进行各种操作。

```
%获取新的树
newt = plot(t,'read',fig);
%从命令行修改新树
newt = wpjoin(newt,3);
%并显示
fig2 = plot(newt);
```

结果如图 12.4 所示，右边显示的是节点 0 处的信号波形。

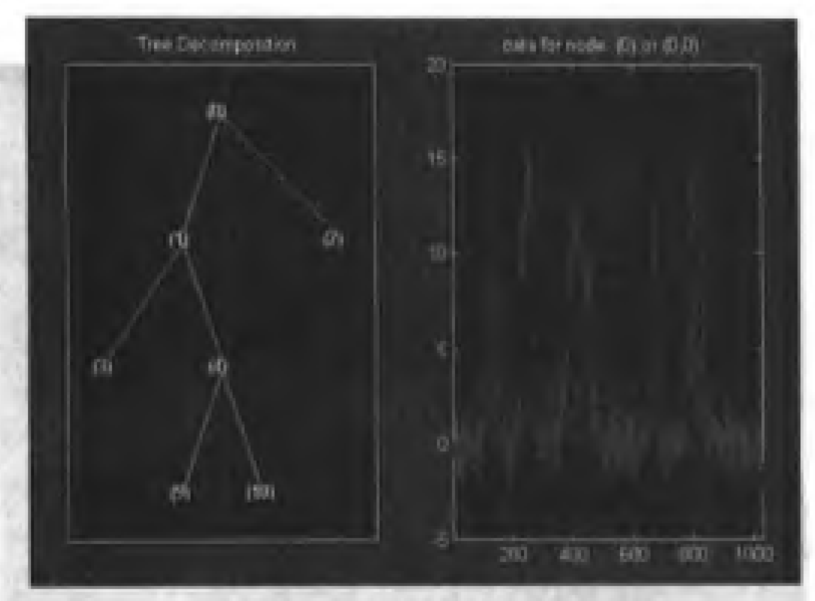


图 12.4　新小波包树的结构

```
%察看颜色表示的小波包系数
wpviewcf(t,1)
```

结果如图 12.5 所示。

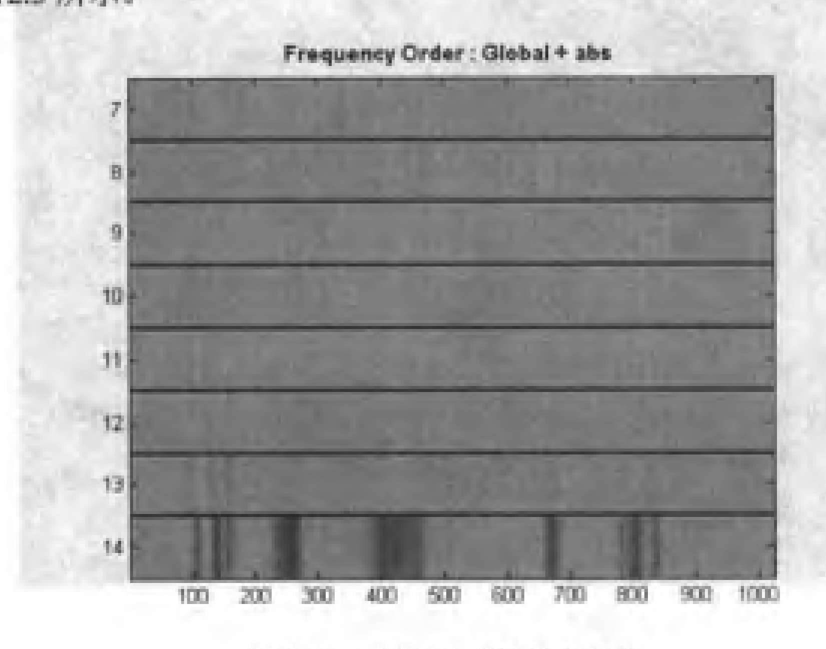


图 12.5　颜色表示的小波包系数

12.2.2　从图形用户接口（GUI）绘制小波包树

例程 12-2：对信号进行小波包分解后，绘制其树结构。

f12_2.m

```
%读信号
load noisbump
x = noisbump;
%3 层小波分解
```

```
t = wpdec(x,3,'db2');
%绘制树结构
fig = drawtree(t);
```

得到图形用户接口界面如图 12.6 所示。

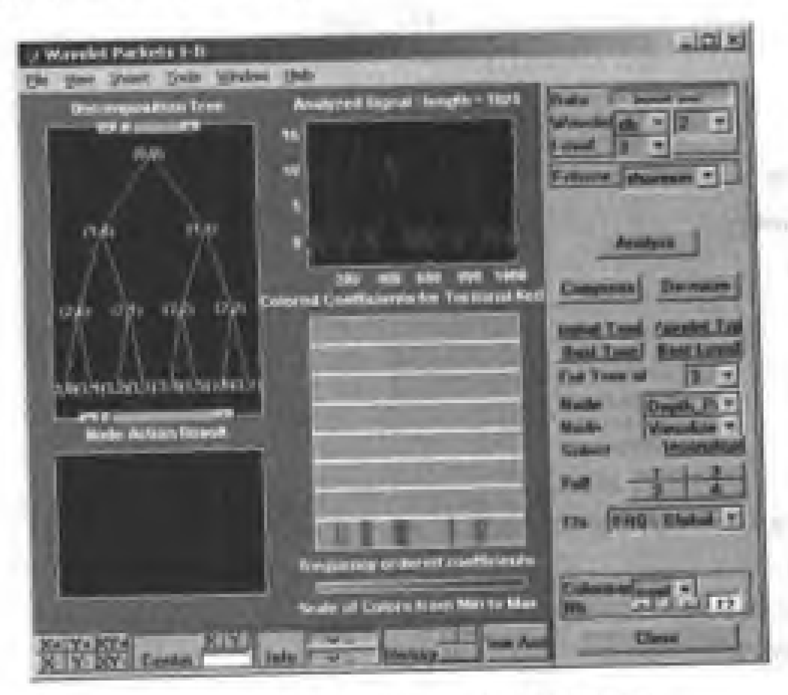


图 12.6　小波包树管理图形用户接口界面

利用这个用户接口界面可以对树结构进行修改。比如将选项“Node label”由“Depth_Position”改为“Index”，将选项“Node Action”由“Visualize”改为“Split/Merge”，然后合并节点 2，得到结果如图 12.7 所示。

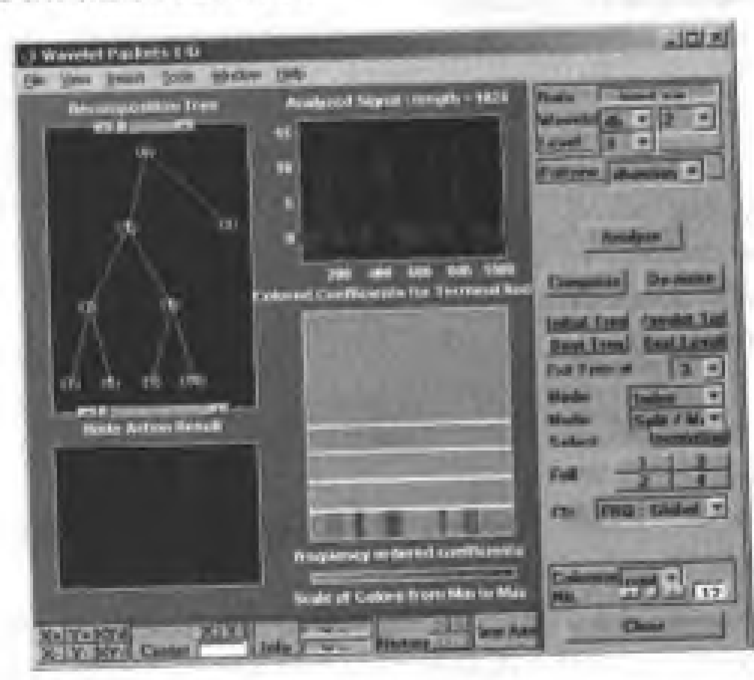


图 12.7　合并节点 2

12.2.3 对小波包设置阈值

例程12-3：对小波包设置阈值。

f12_3.m

```
%读信号
load noisbloc
x = noisbloc;
%3层小波包分解
t = wpdec(x,3,'sym4');
%%显示树结构
plot(t);

%设置全局阈值.
t1 = t;
sorh = 'h';
thr = wthrmngr('wp1ddenoGBL','penalhi',t);
cfs = read(t,'data');
cfs = wthresh(cfs,sorh,thr);
t1   = write(t1,'data',cfs);
plot(t1)

%对每个节点分别设置阈值.
t2 = t;
sorh = 's';
thr(1) = wthrmngr('wp1ddenoGBL','penalhi',t);
thr(2) = wthrmngr('wp1ddenoGBL','sqtwologswn',t);
tn   = leaves(t);
for k=1:length(tn)
   node = tn(k);
   cfs = read(t,'data',node);
   numthr = rem(node,2)+1;
   cfs = wthresh(cfs,sorh,thr(numthr));
   t2 = write(t2,'data',node,cfs);
end
plot(t2)
```

在菜单“Node Label”下选择“Index”，然后点击左边的节点“(0)”，显示其信号波形，得到的结果如图12.8所示。

设置全局阈值后，同样点击节点“(0)”显示其信号波形,如图12.9所示。

对每个节点设置阈值后，同样点击节点“(0)”显示其信号波形，如图12.10所示。

比较可见，经过阈值处理后，信号的波形更加光滑了。

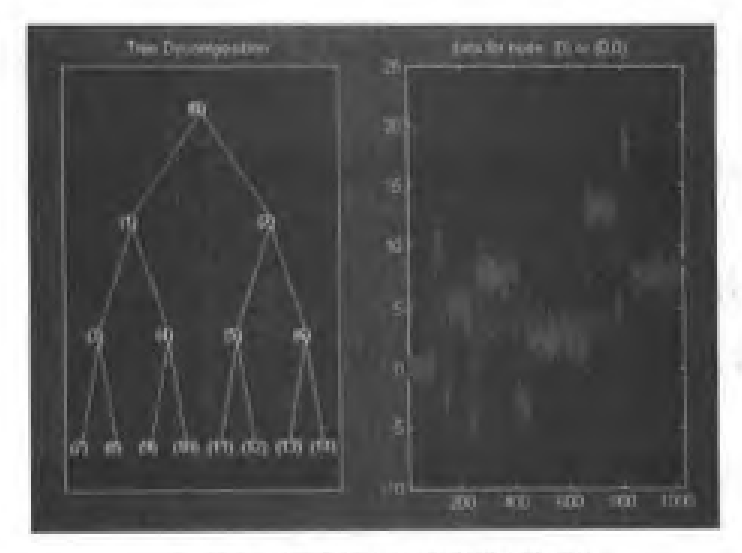


图 12.8 初始节点 0 处的信号波形

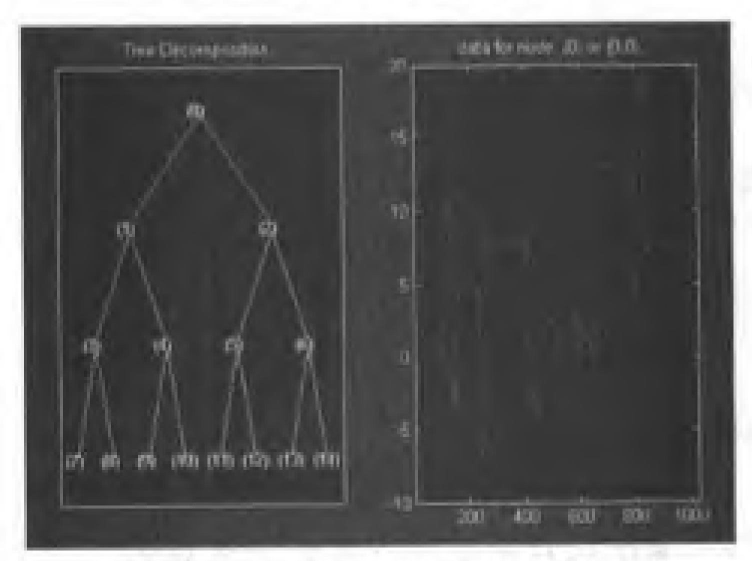


图 12.9 设置全局阈值后节点 0 处的信号波形

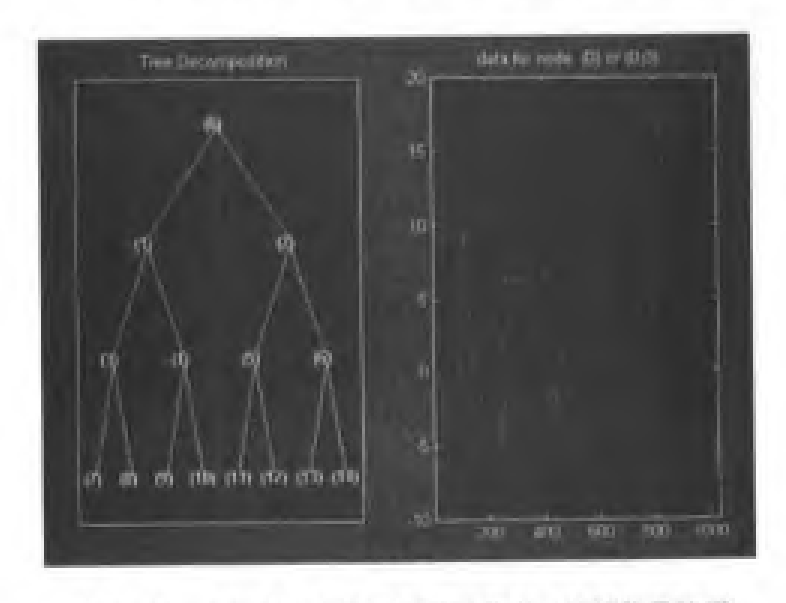


图 12.10 对每个节点设置阈值后节点 0 处的信号波形

12.3 小波包在信号时频分析中的应用

小波包在时频分析中具有明显的优点，下面给出几个典型信号分别在小波基及小波包的最优基下展开后的相平面情况。

12.3.1 δ信号的小波包分析

例程 12-4：计算δ信号小波包变换的时频面。

f12_4.m

```
%delta 信号
x=zeros(1,256);
x(150)=1;
figure(1);
plot(1:length(x),x);
xlabel('样本序号 n');
ylabel('幅值 A');
% 3 层 db1 小波包
wpt = wpdec(x,3,'db1');
%最优基
T=besttree(wpt);
% 小波包结构
plot(wpt);
%db1 小波基下的时频面
wpviewcf(wpt,1);
```

生成的冲击信号波形如图 12.11 所示，它在样本点 150 处存在一个单位冲击。对其进行 3 层 db1 小波进行小波包分解，小波包树结构如图 12.12 所示。

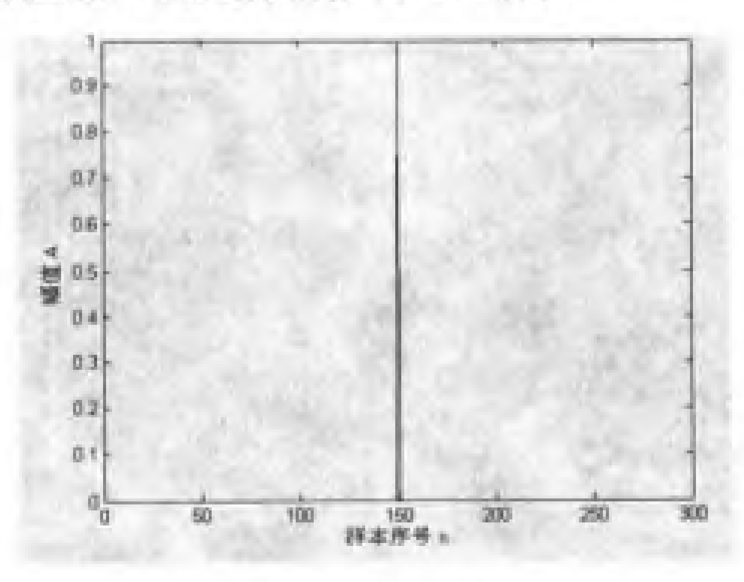


图 12.11 δ 函数的波形

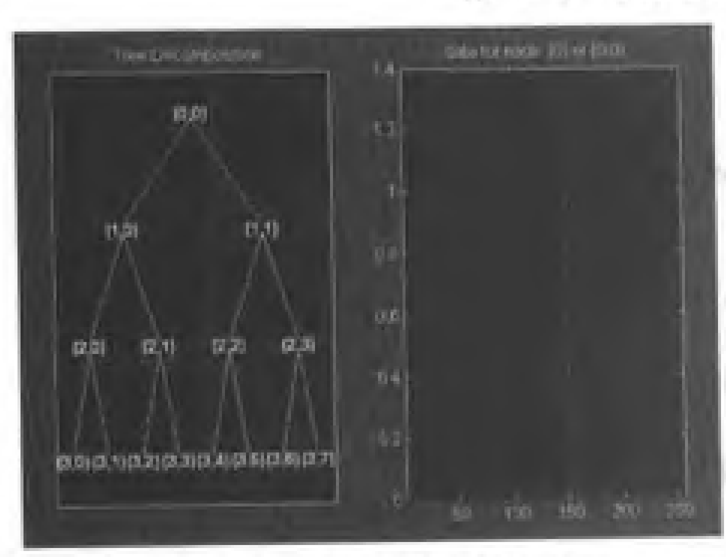


图 12.12　3 层 db1 小波包分解的树结构

分解后的时频相平面如图 12.13 所示，可见能够正确地确定冲击脉冲位置信息。

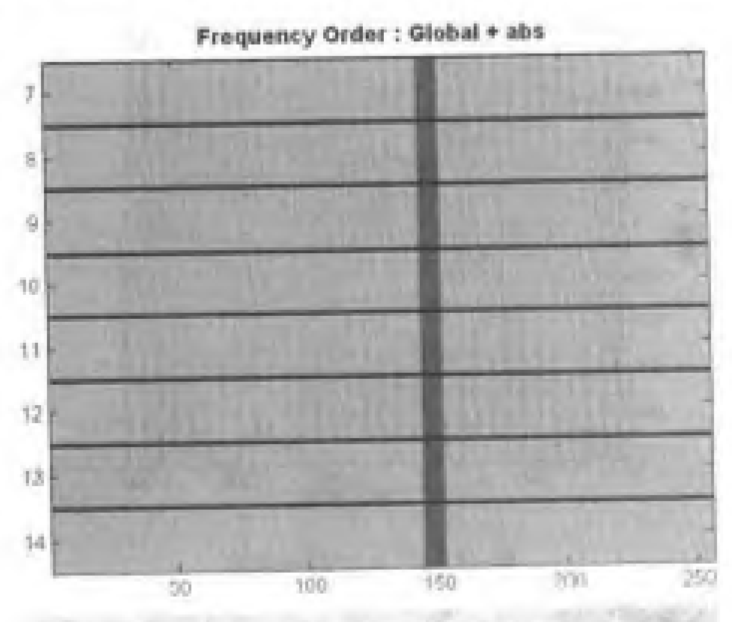


图 12.13　db1 小波分解的时频面

12.3.2　正弦信号的小波包分析

例程 12-5：计算正弦信号小波包变换的时频面。

f12_5.m

```
%读正弦信号
load sinper8;
x = sinper8;
figure(1);
plot(1:length(x),x);
xlabel('样本序号 n');
```

```
ylabel('幅值 A');
% 7层 harr 小波包
% Shannon 熵.
wpt = wpdec(x,7,'haar','shannon');
%最优基
T=besttree(wpt);
% 小波包结构
plot(wpt);
%harr 小波基下的相平面
wpviewcf(wpt,1);
%最优基下的相平面
wpviewcf(T,1);
```

正弦信号的波形如图 12.14 所示，周期为 8，样本长度为 256。对其进行 7 层 haar 小波进行小波包分解，小波包树结构如图 12.15 所示。

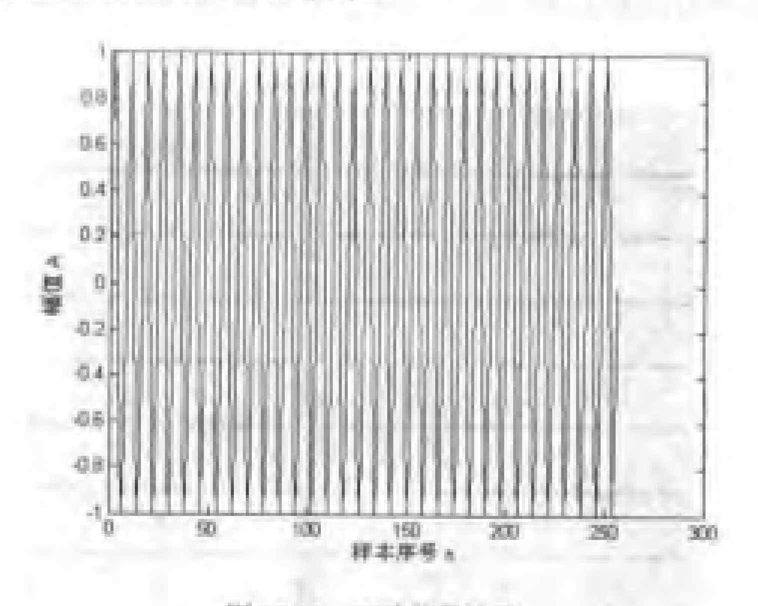


图 12.14　正弦信号波形

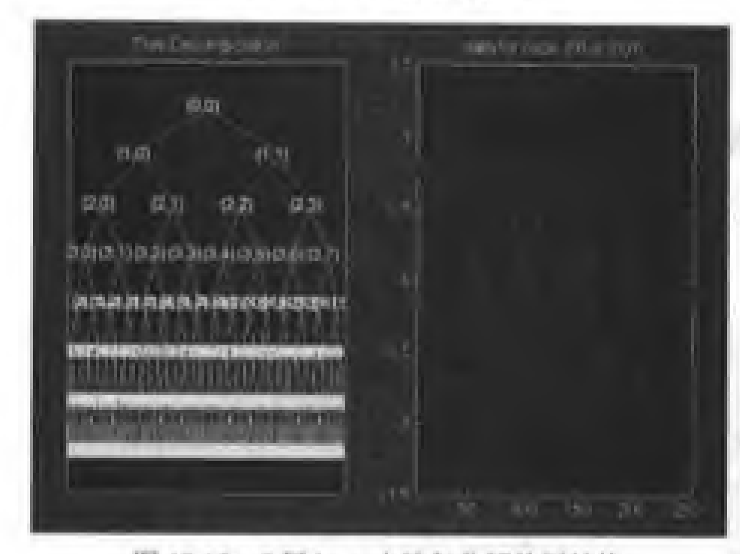


图 12.15　7 层 haar 小波包分解的树结构

haar 小波分解后的时频相平面如图 12.16 所示，最优基分解后的时频相平面如图 12.17 所示，可以看出其频率是一条直线。

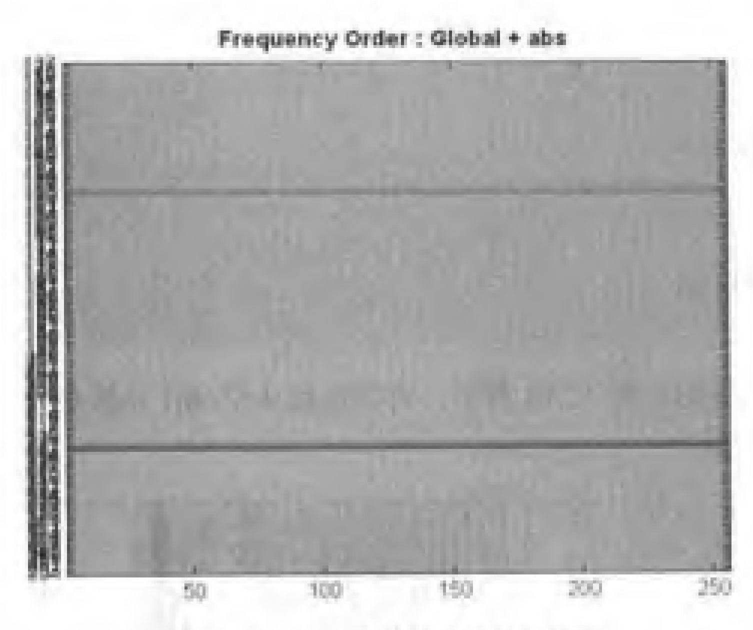


图 12.16 haar 小波基分解的时频面

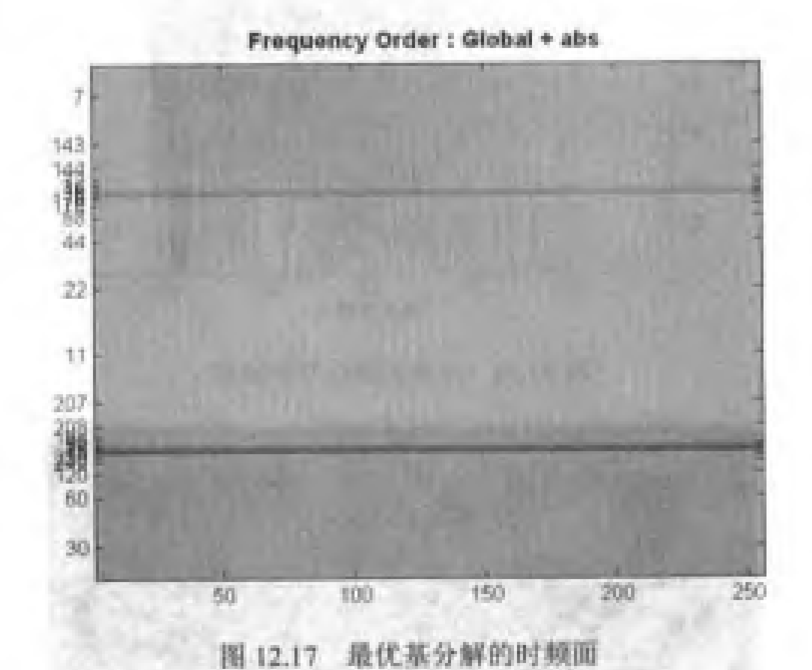


图 12.17 最优基分解的时频面

12.3.3 变频信号的小波包分析

变频信号的表达式为 $\sin(250\pi t^2)(0<t<1)$，采样点数为 512。

例程 12-6：计算变频信号小波包变换的时频面。

f12_6.m

```
%调频信号
load quachirp
```

```
x=quachirp;
figure(1);
plot(1:length(x),x);
xlabel('样本序号 n');
ylabel('幅值 A');
% 4层 db1 小波包
wpt = wpdec(x,4,'db1');
% 小波包结构
plot(wpt);
% db1 小波基下的相平面
wpviewcf(wpt,1);
```

线性调频信号的波形如图 12.18 所示，对其进行 4 层 db1 小波进行小波包分解，小波包树结构如图 12.19 所示。

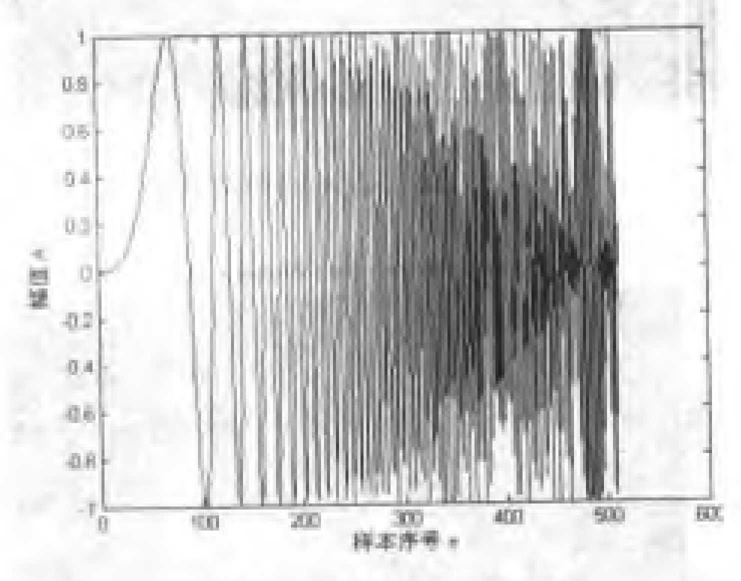


图 12.18 线性调频信号的波形

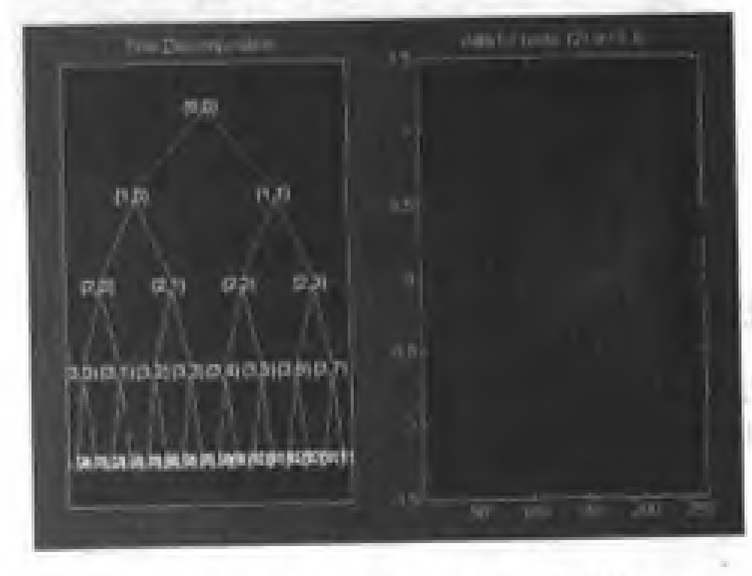


图 12.19 4 层 db1 小波包分解的树结构

db1 小波分解后的时频相平面如图 12.20 所示，可以看出其频率是随时间线性变化的，这与理论结果是一致的。

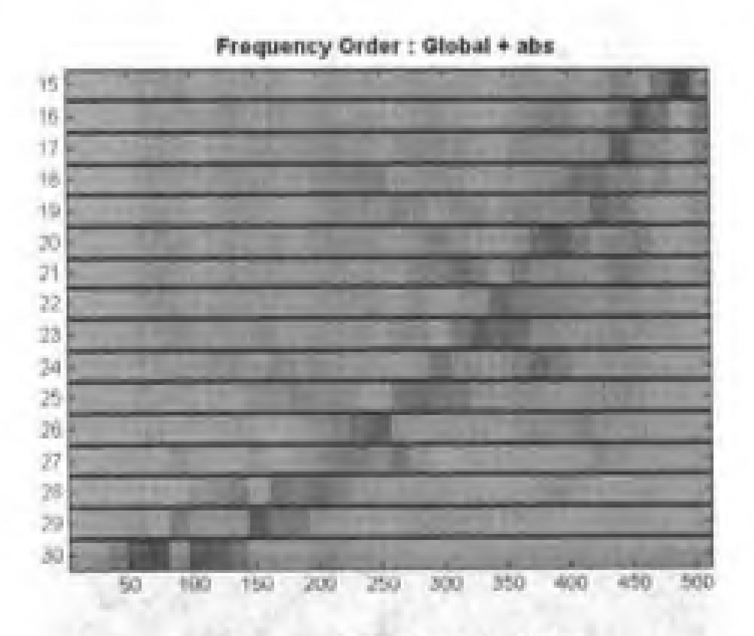


图 12.20　线性调频信号的时频面

12.3.4　调频信号的小波包分析

调频信号的表达式为 $\sin(500\pi t+20\sin 20\pi t)$，采样频率为 1 000Hz。

例程 12-7：计算调频信号的小波包变换时频面。

f12_7.m

```
%调频信号
for i=1:512
    x(i)=sin(500*pi*i/1000+20*sin(20*pi*i/1000));
end
figure(1);
plot(1:length(x),x);
xlabel('样本序号 n');
ylabel('幅值 A');
% 3 层 haar 小波包
wpt = wpdec(x,3,'haar');
%最优基
T=besttree(wpt);
% 小波包结构
plot(wpt);
%haar 小波基下的相平面
wpviewcf(wpt,1);
```

变频信号的波形如图 12.21 所示，对其进行 3 层 haar 小波进行小波包分解，小波包树结构如图 12.22 所示，分解后的时频面如图 12.23 所示，可以看出频率与时间的关系为一个正弦

曲线，这与理论结果是一致的。

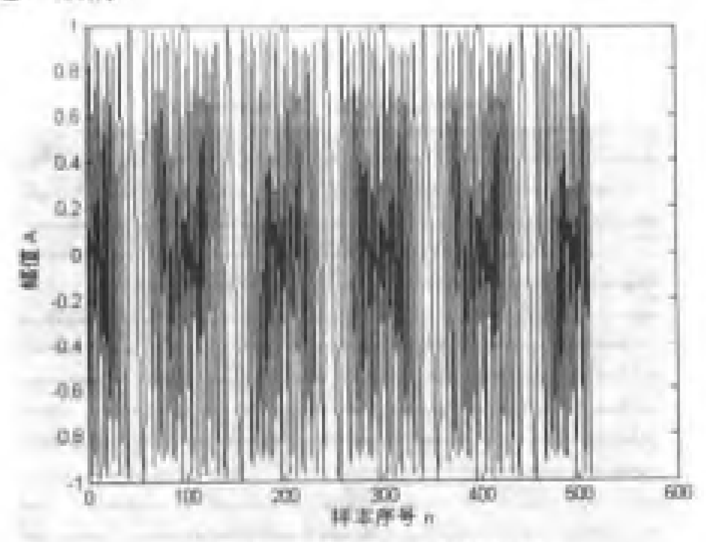


图 12.21　调频信号的时域波形

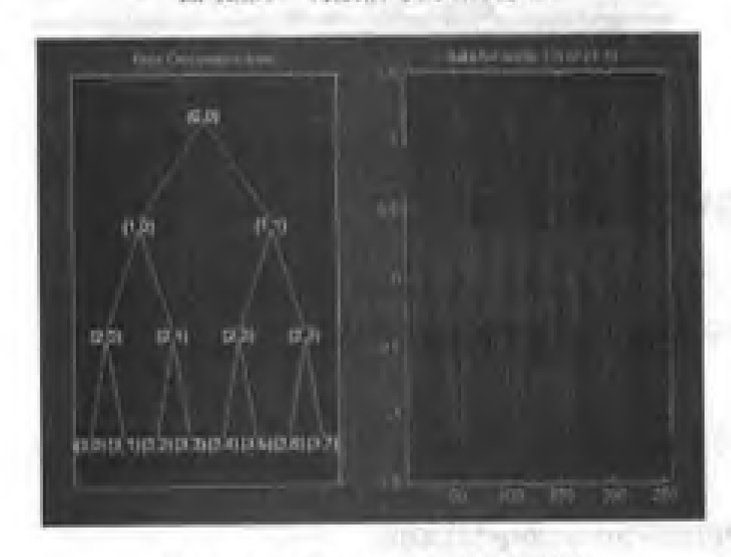


图 12.22　3 层 haar 小波包分解结构

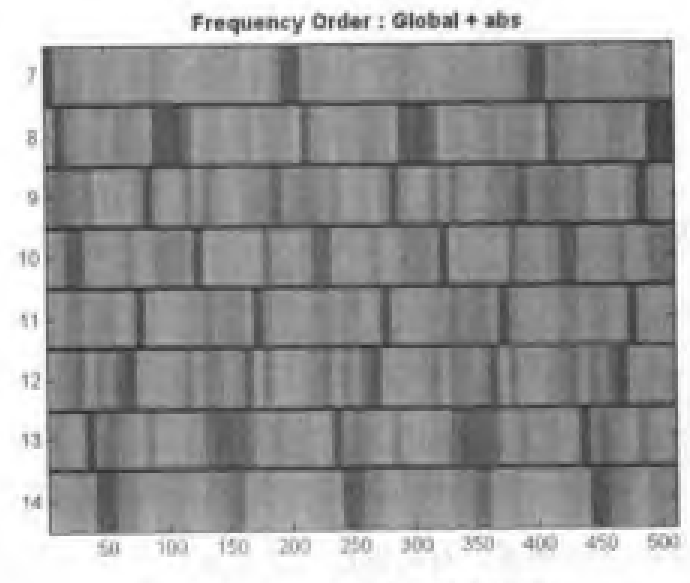


图 12.23　haar 小波包分解的时频面

12.4　小波包在信号消噪中的应用

12.4.1　基本原理

在小波包分析中，其信号消噪的算法思想和在小波分析中的基本相同，所不同的就是小波包提供了一种更为复杂，也更为灵活的分析手段。因为小波包分析对上一层的低频部分和高频部分同时进行分解，具有更加精确的局部分析能力。

对信号进行小波包分解时，可以采用多种小波包基。通常根据分析信号的要求，从中选择最好的一种小波包基，即最优基。最优基的选择标准是熵标准。在 Matlab 的小波工具箱中可通过 besttree 函数进行最优基的选择，即计算最优树。

应用小波包分析对信号进行消噪处理是它的一个最基本的功能。一般地，按照如下步骤进行：

（1）信号的小波包分解。选择一个小波并确定所需分解的层次，然后对信号进行小波包分解。

（2）确定最优小波包基。对于一个给定的熵标准，计算最优树（这一步不是必需的步骤，可根据不同的目的进行有选择性的使用）。

（3）小波包分解系数的阈值量化。对于每一个小波包分解系数，选择一个恰当的阈值并对系数进行阈值量化。

（4）信号的小波包重构。根据最低层的小波包分解系数和经过量化处理系数，进行小波包重构。

在上述的各步中，最关键的是如何选取阈值和如何进行阈值量化，在一定程度上，它直接关系到对信号进行消噪处理的质量。

12.4.2　例程分析

例程 12-8：利用小波包分析对一个给定的含噪信号进行消噪处理。

f12_8.m

```
%装载原始信号并图示之
load noismima;
s=noismima(1:1000);
figure(1);
subplot(2,2,1);
plot(s);
xlabel('样本序号 n');
ylabel('幅值 A');
title('原始信号');

%采用默认阈值，用 wdencmp 函数进行消噪处理
[thr,sorh,keepapp,crit]=ddencmp('den','wp',s);
```

```
%用全局阈值选项进行消噪处理
[c,treed,perf0,perfl2]= wpdencmp(s,sorh,3,'db2',crit,thr,keepapp);
subplot(2,2,3);
plot(c);
xlabel('样本序号 n');
ylabel('幅值 A');
title('默认阈值消噪信号');

%根据前面的消噪效果，调节阈值大小进行消噪
thr=thr+15;
[c1,treed,perf0,perfl2]= wpdencmp(s,sorh,3,'db2',crit,thr,keepapp);
subplot(2,2,4);
plot(c1);
xlabel('样本序号 n');
ylabel('幅值 A');
title('调节阈值后的消噪信号');
```

结果如图 12.24 所示，可以明显看出，小波包很好地消除了原始信号中的噪声。

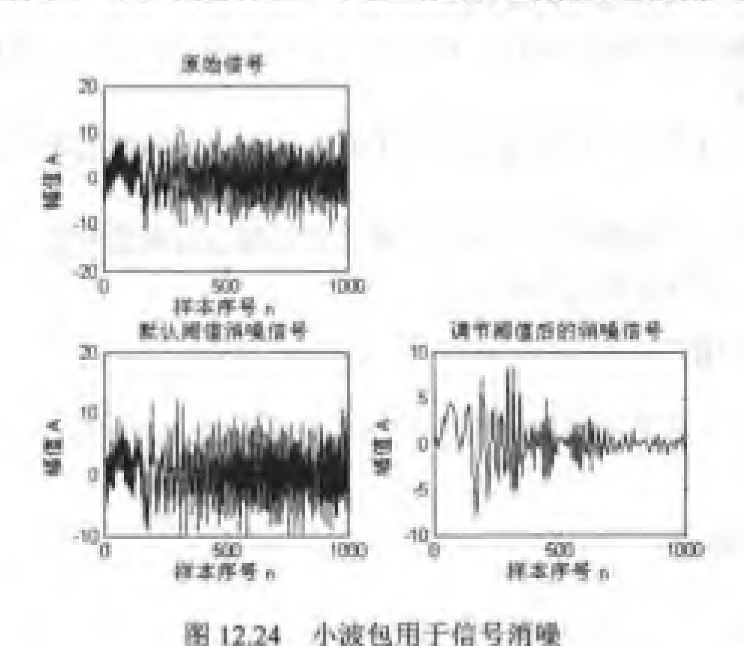


图 12.24　小波包用于信号消噪

12.5　小波包在信号压缩中的应用

12.5.1　基本原理

在小波包分析中，其信号压缩的算法思想和在小波分析中的基本相同，所不同的就是小波包提供了一种更为复杂，也更为灵活的分析手段。因为小波包分析对上一层的低频部分和

高频部分同时进行分解，具有更加精确的局部分析能力。

在用小波包分析进行信号压缩处理中，最关键的是如何选取阈值和如何进行阈值量化，在一定程度上，这直接关系到对信号进行压缩处理的质量。

12.5.2 例程分析

例程 12-9：利用小波包分析对一个给定的一维信号进行压缩处理。

f12_9.m

```
%装载源信号
load noisbump;
s=noisbump(1:1000);
figure(1);
subplot(2,1,1);
plot(s);
xlabel('样本序号 n');
ylabel('幅值 A');
title('原始信号');
%采用默认阈值，以小波包函数 wpdencmp 对 s 进行压缩处理
[thr,sorh,keepapp,crit]=ddencmp('cmp','wp',s);
[sc,treed,perf0,perfl2]= wpdencmp(s,sorh,3,'db2',crit,thr,keepapp);
subplot(2,1,2);
plot(sc);
xlabel('样本序号 n');
ylabel('幅值 A');
title('压缩后的信号');
```

结果如图 12.25 所示。

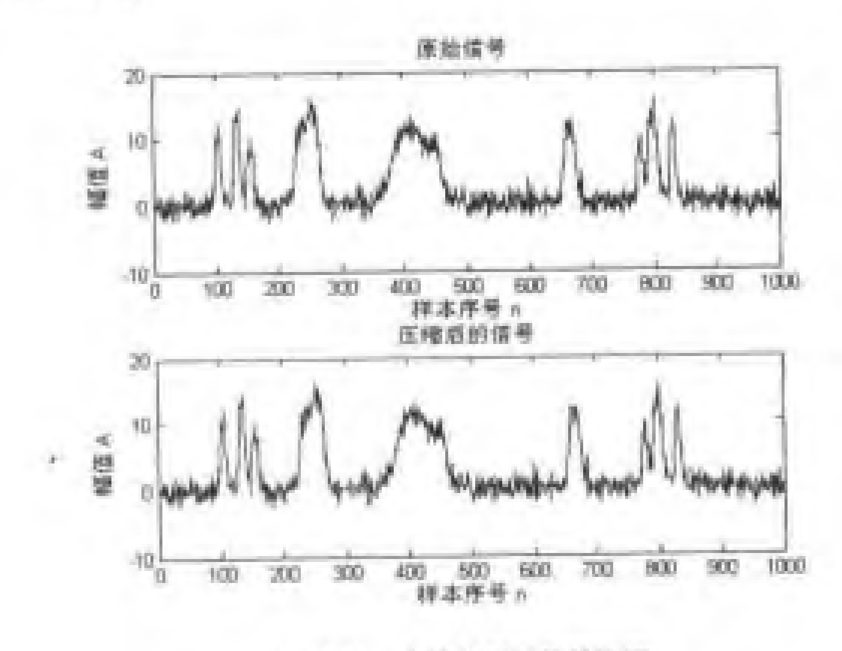


图 12.25　小波包用于信号压缩

```
perf0 =
    38.7352
perfl2 =
    99.0404
```

这说明上述压缩过程保留了 99.04%的能量，压缩率 38.73%。

第 13 章　小波包在图像处理中的应用

相对于小波变换，小波包变换能够对图像中的高频部分进行分解，具有更强的适应性，因此更加适合于图像的各种处理。

本章主要内容包括：

- 小波包在图像消噪中的应用；
- 小波包在图像压缩中的应用；
- 小波包在图像边缘检测中的应用。

13.1　引　　言

图像的多分辨率分析是一种在不同的分辨率下处理图像中不同信息的方法，它将图像在各种分辨率下的细节提取出来，得到一个拥有不同分辨率的图像细节序列后再进行图像的各种处理。

小波包分解与小波分解相比，是一种更精细的分解方法，它不仅对图像的低频部分进行分解，也要对图像的高频部分进行分解。小波包对图像分解作多分辨率分解是在小波函数对图像的分解基础上发展起来的，通过水平和垂直滤波，小波包变换将原始图像分为 4 个子带：水平和垂直方向上的低频子带，水平和垂直方向上的高频子带。继续对图像的低频子带和高频子带进行分解就可以得到图像的小波包分解树结构，如图 13.1 所示。

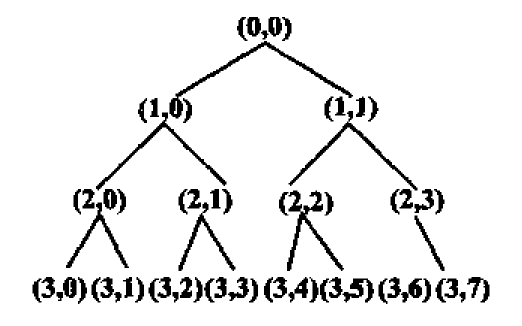


图 13.1　图像的小波包分解结构示意图

由图 13.1 可见，分解级数越大，也就是选择的小波包尺度越大，小波包系数对应的空间分辨率就越低，利用这一点，可以在不同的空间分辨率上进行分析，实现图像的消噪、压缩、编码等各种处理工作。

13.2　小波包在图像消噪中的应用

13.2.1　基本原理

在本节中，小波包分析进行图像消噪处理的基本原理和方法与前一章中所介绍的对信号消噪的相同，请读者参看第 12 章的相关内容。本节仅以具体的实例来说明小波包分析在图像消噪处理中的应用。

在 Matlab 7.0 的小波工具箱中，提供了一个函数 wpdencmp，它是专门利用小波包分解实现消噪和压缩处理的。

wpdencmp 函数

【语法格式】

① [XD,TREED,PERF0,PERFL2] =

wpdencmp(X,SORH,N,'wname',CRIT,PAR,KEEPAPP)

② [XD,TREED,PERF0,PERFL2] =

wpdencmp(TREE,SORH,CRIT,PAR,KEEPAPP)

【使用说明】利用小波包实现信号和图像的消噪或压缩。

输入参数：sorh 指定选取软阈值（sorh='s'）或硬阈值（sorh='h'）；N 为小波分解的层数；wavename 指定分解时所用的小波；CRIT 和 PAR 定义了熵准则；TREE 是小波包分解树结构。

输出参数：perfo,perfl2 返回压缩比例系数。

13.2.2 例程分析

例程 13-1：利用小波包变换对一个二维含噪图像进行消噪处理。

f13_1.m

```
%装载并显示原始图像
load gatlin2;
subplot(2,2,1);
image(X);
colormap(map);
title('原始图像');
axis square;

%在图像中加入噪声
init=2055615866;
randn('seed',init);
X1=X+10*randn(size(X));
subplot(2,2,2);
image(X1);
colormap(map);
title('含噪图像');
axis square;

%基于小波包的消噪处理
thr=10;sorh='s';
crit='shannon';
keepapp=0;
X2=wpdencmp(X1,sorh,3,'sym4',crit,thr,keepapp);
```

```
%画出消噪后的图像
subplot(2,2,3);
image(X2);
colormap(map);
title('全局阈值消噪图像');
axis square;

%对图像进行平滑处理以增强消噪效果（中值滤波）
for i=2:175;
    for j=2:259
        Xtemp=0;
        for m=1:3
            for n=1:3
                Xtemp=Xtemp+X2(i+m-2,j+n-2);
            end
        end
        Xtemp=Xtemp/9;
        X3(i,j)=Xtemp;
    end
end

%显示平滑结果
subplot(2,2,4);
image(X3);
colormap(map);
title('平滑后的图像');
axis square;
```

计算结果如图 13.2 所示，其中左上角是原始的图像，右上角是添加噪声后的图像，通过小波包分解并设置全局阈值小噪后的结果如左下图所示，与含噪图像相比，明显清楚了很多；进一步对消噪后的图像进行平滑处理，如右下图所示，与消噪后的图像相比，它明显光滑了。



图 13.2　基于小波包变换的图像消噪

除了利用函数 wpdencmp 对图像进行消噪外，还可以利用二维小波包分解函数 wpdec2 来实现图像消噪。

wpdec2 函数

【语法格式】

① T = wpdec2(X,N,'wname',E,P)

② T = wpdec2(X,N,'wname')

【使用说明】二维小波包分解

输入参数：X 是分析矩阵；N 是分解的层数；wname 是小波基函数；E 是熵的类型；P 是依赖于 E 的可选参数。

输出参数：T 返回小波包树。

例程 13-2：利用二维小波包分解对一个二维含噪图像进行消噪处理。

f13_2.m

```
%装载并显示原始图像
load flujet;
subplot(2,2,1);
image(X);
colormap(map);
title('原始图像');
axis square;

%生成含噪图像
init=2055615866;
randn('seed',init);
X1=X+20*randn(size(X));
subplot(2,2,2);
image(X1);
colormap(map);
title('含噪图像');
axis square;

%用小波 sym2 对图像 X1 进行一层小波包分解
T=wpdec2(X1,1,'sym2');

%设置阈值
thr=8.342;

%对图像的小波包分解系数进行软阈值量化
NT=wpthcoef(T,0,'s',thr);

%仅对低频系数进行重构
X2=wprcoef(NT,1);

%画出消噪后的图像
subplot(2,2,3);
image(X2);
colormap(map);
title('消噪后的图像');
axis square;
```

计算结果如图 13.3 所示，其中左上角是原始的图像，右上角是添加噪声后的图像，通过二维小波包分解后并对系数进行阈值化处理后，重构的图像如左下图所示，与含噪图像相比，它明显清楚了很多，达到了消噪的效果。

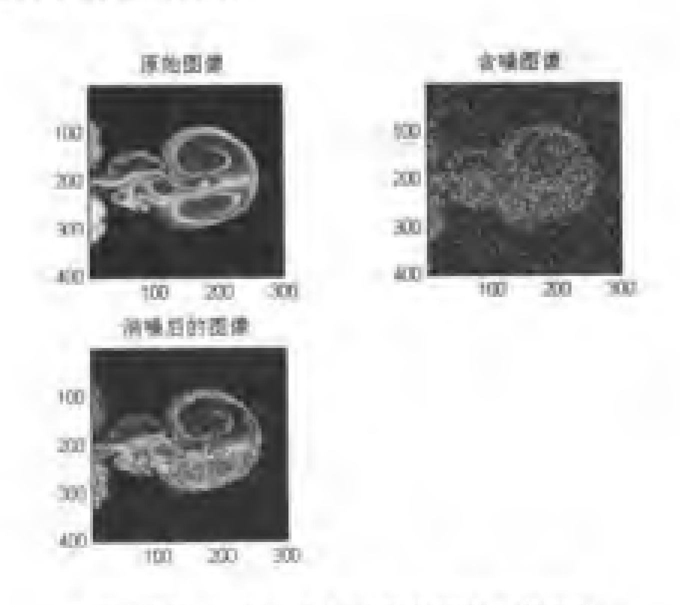


图 13.3　基于二维小波包分解的图像消噪

13.3　小波包在图像压缩中的应用

13.3.1　基本原理

由 Kunt 等人提出的第二代图像数据压缩算法，充分考虑了人类视觉生理心理特征，侧重于将原始图像在频率内作多层分解，然后对这些信息表示灵活地有选择地编码，可得到高的压缩比和很小的失真度，比第一代图像数据压缩算法仅考虑图像本身的效果好。小波包具有能将空间作精细分解的性质，所以很适合于进行图像数据的压缩。

在本节中，小波包分析进行图像压缩处理的基本原理和方法与前一章中所介绍的对信号压缩的相同，请读者参看第 12 章的相关内容。本节仅以具体的实例来说明小波包在图像压缩处理中的应用。

下面通过一个具体的例子来说明小波包的压缩功能。

13.3.2　例程分析

例程 13-3：利用小波包分析对给定图像进行压缩处理。

f13_3.m

```
%装载并显示原始图像
load wbarb;
subplot(1,2,1);
image(X);
```

```
colormap(map);
title('原始图像');

%采用默认的全局阈值
[thr,sorh,keepapp,crit]=ddencmp('cmp','wp',X);
%图像进行压缩
Xc=wpdencmp(X,sorh,3,'bior3.1',crit,thr,keepapp);

%显示压缩结果
subplot(1,2,2);
image(Xc);
colormap(map);
title('全局阈值压缩图像');
```

计算结果如图 13.4 所示，其中左图是原始的图像，右图是经过全局阈值压缩过的图像，比较可见，压缩后的图像基本上保持了原来图像的内容，但其包含的大量冗余信息被剔除了。

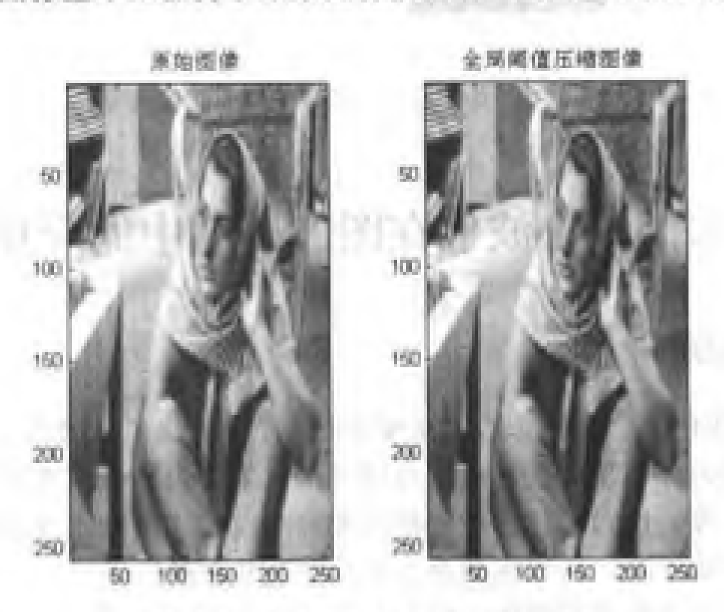


图 13.4 基于小波包分解的图像压缩

例程 13-4：利用小波包进行图像压缩处理。

f13_4.m

```
%装载并显示原始图像
load detfingr;
subplot(1,2,1);
image(X);
colormap(map);
title('原始图像');

%采用默认的全局阈值
[thr,sorh,keepapp,crit]=ddencmp('cmp','wp',X);
```

```
%图像压缩
[Xc,treed,perf0,perfl2]=wpdencmp(X,sorh,3,'bior3.1',crit,thr,keepapp);
    subplot(1,2,2);
    image(Xc);
    colormap(map);
    title('压缩后的图像');

    %给出压缩效率
    disp('小波分解系数中置 0 的系数个数百分比：');
    perf0
    disp('压缩后图像剩余能量百分比：');
    perfl2
```

计算结果如图 13.5 所示，其中左图是原始的图像，右图是经过全局阈值压缩过的图像。小波分解系数中置 0 的系数个数百分比为

perf0 =

    58.0203

压缩后图像剩余能量百分比为

perfl2 =

    99.9778

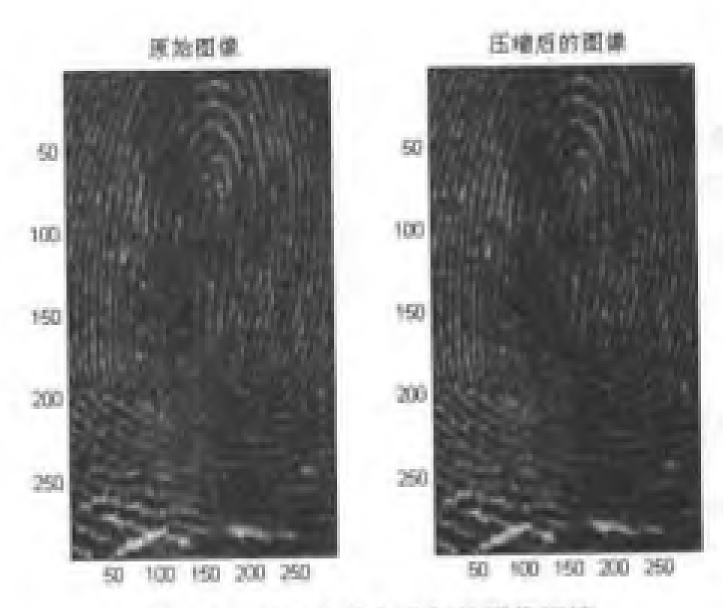


图 13.5　基于小波包分解的图像压缩

13.4　小波包在图像边缘检测中的应用

13.4.1　基本原理

图像的边缘检测是对图像进行进一步处理和识别的基础，虽然图像边缘产生的原因不同，

但反映在图像的组成基元上，它们都是图像上灰度的不连续点或灰度剧烈变化的地方，这就意味着图像的边缘就是信号的高频部分。因此所有的边缘检测方法都是检测信号的高频分量，但是在实际图像中，由于噪声的存在，边缘检测成为一个难题。

小波包分解后得到的图像序列有近似部分和细节部分组成，近似部分是原图像对高频部分进行滤波所得的近似表示。经滤波后，近似部分去除了高频分量，因此能够检测到原图像中所检测不到的边缘。

13.4.2 例程分析

例程 13-5：利用小波包分解检测图像的边缘。

f13_5.m

```
%装载并显示原始图像
load woman;
%加入含噪
init=2055615866;
randn('seed',init);
X1=X+20*randn(size(X));
figure(1);
image(X1);
colormap(map);
title('原始图像');
axis square;

%用小波 db4 对图像 X 进行一层小波包分解
T=wpdec2(X1,1,'db4');

%重构图像近似部分
figure(2);
A=wprcoef(T,[1 0]);
image(A);
title('图像的近似部分');
axis square;

%边缘检测
figure(3);
%%原图像的边缘检测
BW1 = edge(A,'sobel');
imshow(BW1);
title('原图像的边缘');
axis square;

%%图像近似部分的边缘检测
figure(4);
BW2= edge(X1,'sobel');
```

```
imshow(BW2);
title('图像近似部分的边缘');
axis square;
```

原始的含噪图像如图 13.6 所示，利用 db4 正交小波基对其进行一层小波包分解后，重构其近似部分如图 13.7 所示，比较可见，经小波包分解后所得到的近似图像与原图像层次更加分明，因此利用分解后的近似图像能检测边缘。图 13.8 是直接对原图像进行边缘检测的结果，图 13.9 是对近似图像进行边缘检测的结果，比较可见，后一种方法的效果更好。



图 13.6　原始的含噪图像



图 13.7　图像小波包分解后的近似部分



图 13.8　原图像的边缘检测



图 13.9　近似图像的边缘检测

第 14 章　小波分析在交通领域中的应用

目前小波分析在交通领域中的应用已经引起了广泛注意，许多学者投入到这方面的研究。小波分析在交通领域中应用包括有路面不平度分析、车辆的振动分析等。特别是小波分析非常适合于分析非平稳信号，因此小波分析可作为车辆故障诊断中信号处理的较理想工具，由它可以构造故障诊断所需的特征或直接提取对诊断有用的信息。

本章的主要内容包括：

- 小波变换在汽车变速器轴承故障诊断中的应用；
- 小波变换在汽车变速器齿轮故障诊断中的应用；
- 小波变换在路面不平度分析中的应用；
- 小波变换在车辆非平稳振动分析中的应用。

14.1　引　　言

利用小波变换的多分辨率性质，基于信号和随机噪声在小波变换域中不同的模极大值系数特征，不但能提取信号和噪声在多尺度分辨空间中的波形特征，而且根据表征该特征的小波系数模极大值传播特性的不同，来实现对信号波形的有效检测。这样，既避免了矩阵运算，降低了运算量，又能在获得一定改善信噪比增益的同时，保持对信号波形细节有较好分辨率，并且对待检测信号形式不敏感。

传统的信号处理方法大都基于傅立叶变换，因而必然面临傅立叶分析的一对基本矛盾：时域和频域局部化的矛盾，并且傅立叶分析是以信号平稳性假设为前提的，而实际中很多物理过程的信号往往包含在瞬态信号及时变信号中。小波的时频分析方法不仅能够提供信号的全部信息，而且又能提供在任一局部时间内信号变化激烈程度的信息，即可提供时频同时局部化的信息。

小波分析是一种全新的信号的时间-尺度分析方法，它继承了傅立叶分析用简谐函数作为基函数来逼近任意信号思想，只不过小波分析的基函数是一系列尺度可变函数。这使得小波分析具有良好的时-频定位特性以及对信号的自适应能力，故而能够对各种时变信号进行有效地分解，为控制系统故障诊断提供了新的、强有力的分析手段。小波变换属线性变换，无干扰项，它具有多分辨率分析的特点，即时频分辨率可变，具有“变焦”特性，因此具有对非平稳信号局部化分析的突出优点，有良好的时-频定位功能，很适合探测正常信号中夹带的瞬态反常现象并揭示其成分。因此，在非平稳信号处理中，利用小波变换进行分析具有广阔的应用前景。本章将举例阐述小波分析在交通领域中的不同应用。

14.2 小波变换在汽车变速器轴承故障诊断中的应用

变速器是汽车传动系统的重要组成部分，同时又是一种容易损坏的部件。构成变速器的齿轮、轴承和轴工作状况复杂，随着变速器性能的不断完善、结构的进一步复杂，传统的人工手段以及撤卸解体诊断已经不能适应需求。将现代信号处理方法应用于变速器的故障诊断中，并研究在不解体的情况下，有效地检测诊断故障十分重要。传统的频谱方法只适用于分析平稳信号，而变速器中滚动轴承、齿轮出现故障时会产生调幅、调频振动信号，它们与时间有着明显的依赖关系。对这类信号的处理，仅在频域或时域进行分析是不够的，必须要有一种新的方法能够将时域和频域结合起来描述观察信号的时频特征，而小波分析的优点使得小波分析能够很好地解决这类问题。

滚动轴承在工作时，一般是外圈于轴承座或机壳相联接，固定或相对固定；内圈与机械传动轴相联接，随轴一起转动。在机械运转时，由于轴承本身的结构特点、加工装配误差及运行过程中出现的故障等内部因素，当轴以一定的速度并在一定的载荷下运转时对轴承和轴承座组成的振动系统产生激励，使系统产生振动。

滚动轴承在运行过程中出现的故障按其振动信号的特征不同可分为两大类：一类称为表面损伤类故障，如点蚀、剥落、擦伤等；另一类称为磨损故障。

对于表面损伤类故障，当损伤点滚过轴承元件表面时，要产生突变的冲击脉冲力，该脉冲力为一宽带信号，所以必然覆盖轴承系统的各个固有频率从而引起轴承的振动，这就是损伤类故障引起的振动信号的基本特征。同时，这种由表面损伤故障引起的振动响应往往会被较大的振动信号所掩盖，从而无法从功率谱中分辨出来。

小波分析由于具有同时分析信号时域与频域的特性，所以使用小波分析技术对检测的信号进行变换，然后对具有故障特征的信号进行重构，最后通过希尔波特变换进行解调和细化频谱分析，从而轴承中的故障信息成分就可以检测出来，以判断轴承发生故障的部位。

14.2.1 轴承外环故障诊断

轴承故障设置为外环局部剥落，这种情况下，轴承故障特征频率为外环通过频率 f_{op} 及其倍频，理论计算得到 f_{op}=475.6Hz。

例程 14-1：利用小波变换检测轴承外环故障。

f14_1.m

```
%采样频率
fs=10000;
%轴承外环故障信号
fid=fopen('bearingout.dat','r');%故障
N=1024;
xdata=fread(fid,N,'int16');
fclose(fid);
xdata=(xdata-mean(xdata))/std(xdata,1);
```

```
%时域波形
figure(1);
plot(1:N,xdata);
xlabel('样本序号 n/个');
ylabel('电压 V/v');

%db10 小波进行 4 层分解
%一维小波分解
[c,l] = wavedec(xdata,4,'db10');

%重构第 1-4 层细节信号
d4 = wrcoef('d',c,l,'db10',4);
d3 = wrcoef('d',c,l,'db10',3);
d2 = wrcoef('d',c,l,'db10',2);
d1 = wrcoef('d',c,l,'db10',1);

%显示细节信号
figure(2)
subplot(4,1,1);
plot(d4,'LineWidth',2);
ylabel('d4');
subplot(4,1,2);
plot(d3,'LineWidth',2);
ylabel('d3');
subplot(4,1,3);
plot(d2,'LineWidth',2);
ylabel('d2');
subplot(4,1,4);
plot(d1,'LineWidth',2);
ylabel('d1');
xlabel('样本序号 n/个');

%第 1 层细节信号的包络谱
y=hilbert(d1);
ydata=abs(y);
y=y-mean(y);
nfft=1024;
p=abs(fft(ydata,nfft));
figure(3);
plot((0:nfft/2-1)/nfft*fs,p(1:nfft/2));
xlabel('频率 f/Hz');
ylabel('功率谱 P/W');
```

采集的外环故障振动信号时域波形如图 14.1 所示，从波形图上根本看不出是否存在外环故障。

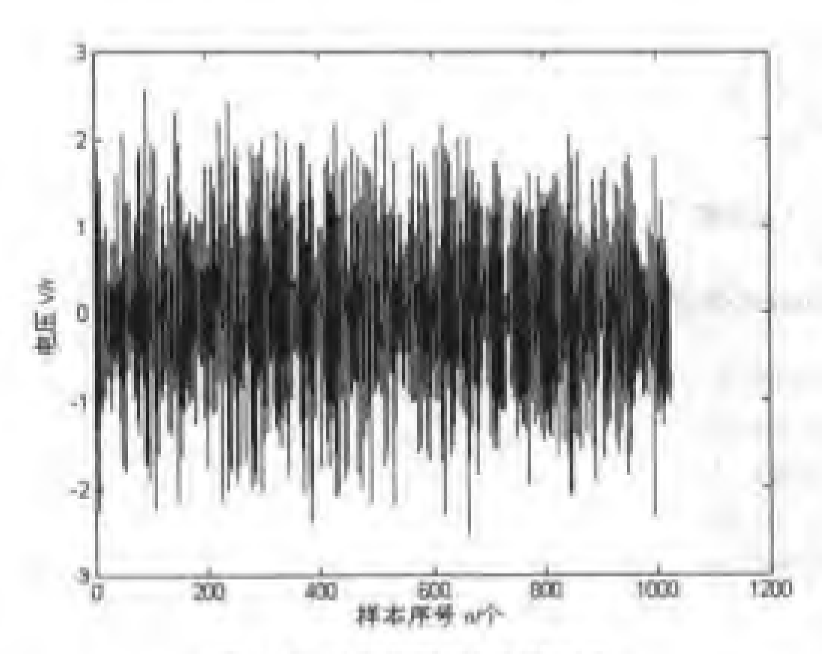


图 14.1 外环故障信号时域波形

对图 14.1 中的信号用 db10 正交小波基进行 4 层小波分解，分解结果如图 14.2 所示，其中 d1～d4 分别表示第一、二、三、四层细节信号。为了进一步提取外环故障特征频率，对第一层细节信号 d1 作 Hilbert 包络并进行谱分析，结果如图 14.3 所示。从功率谱的分析中可以发现频率 475Hz 的存在，通过对照轴承故障特征频率可知，轴承的外环发生了故障。

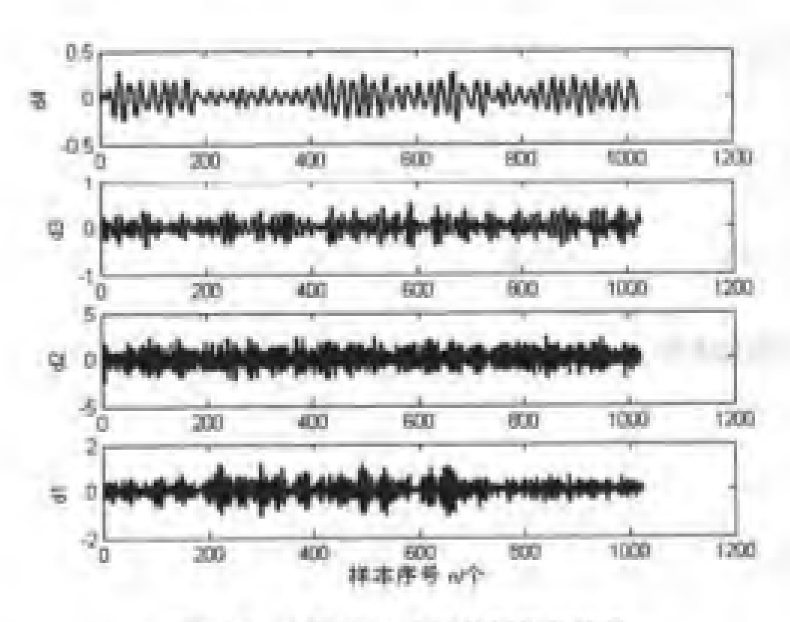


图 14.2 4 层 db10 小波分解细节信号

通常根据小波的正交性质，可将其分为正交小波、半正交小波、双正交小波和非正交小

波，不同的小波具有不同的性质和一定的适用范围，但目前对小波的选择尚缺乏完善的理论加以指导，因而给小波应用带来一定困难。为了比较不同正交小波基对诊断性能的影响，分别利用 db1 和 db5 正交基对上述数据进行了同样的分析。

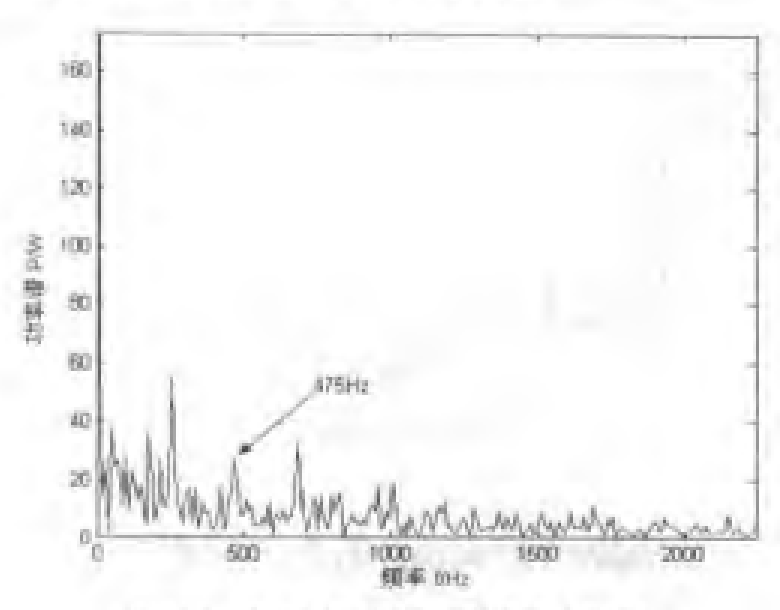


图 14.3 第 1 层细节信号的包络谱（db10）

利用 db1 正交小波基进行 4 层小波分解的结果如图 14.4 所示，其中 d1～d4 分别表示第一、二、三、四层细节信号，对其第一层细节信号 d1 作 Hilbert 包络并进行谱分析，结果如图 14.5 所示。利用 db5 正交小波基进行 4 层小波分解的结果如图 14.6 所示，其中 d1～d4 分别表示第一、二、三、四层细节信号，对其第一层细节信号 d1 作 Hilbert 包络并进行谱分析，结果如图 14.7 所示。比较可见，利用 db1 正交小波基得到的故障频率最明显。

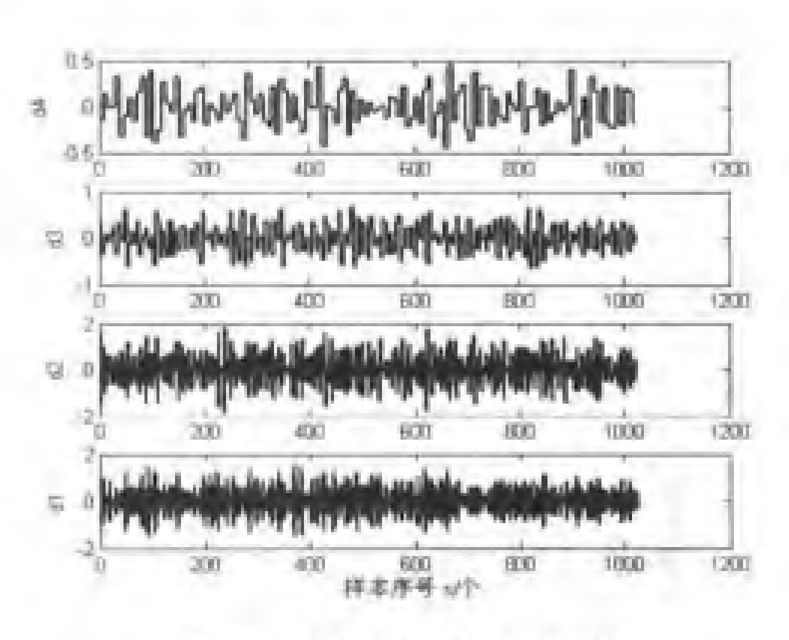


图 14.4 4 层 db1 小波分解细节信号

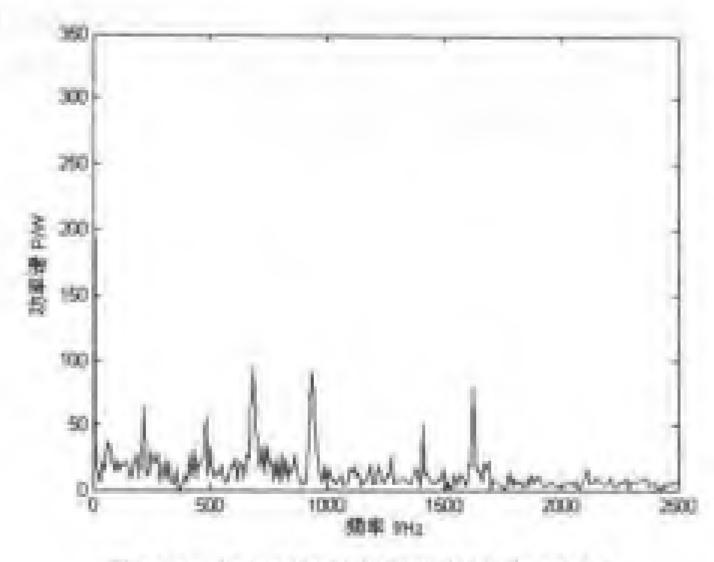


图 14.5　第 1 层细节信号的包络谱（db1）

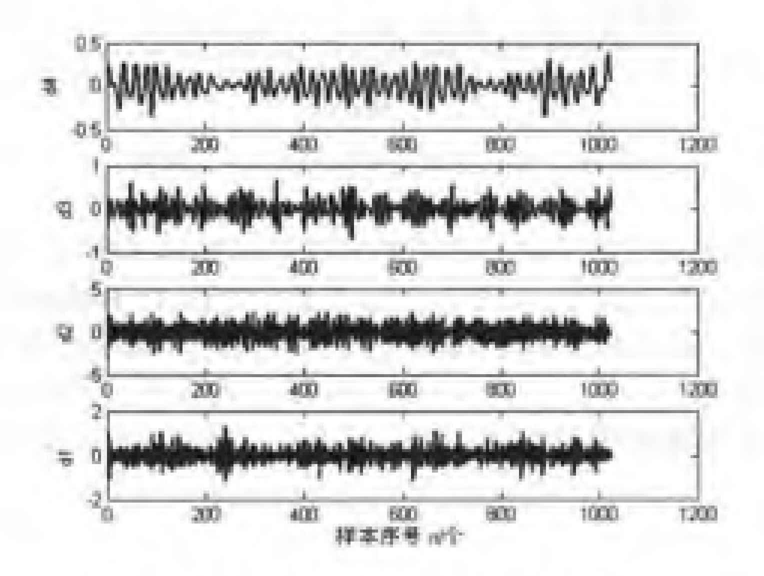


图 14.6　4 层 db5 小波分解细节信号

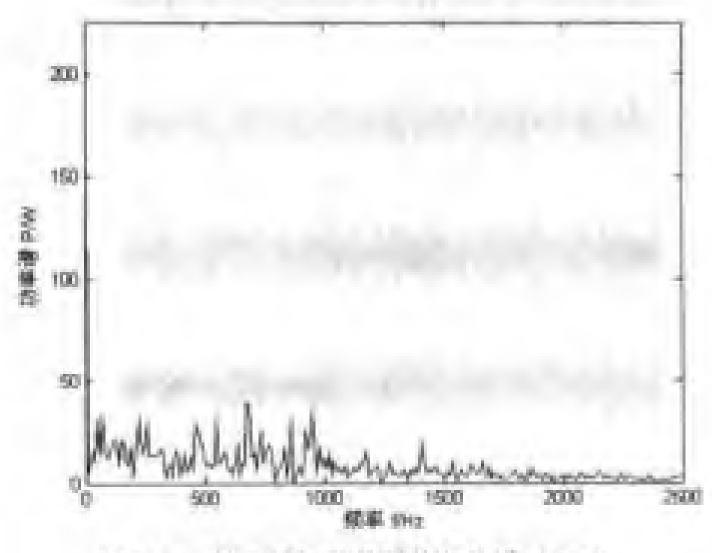


图 14.7　第 1 层细节信号的包络谱（db5）

14.2.2　轴承滚动体故障诊断

轴承故障设置为滚动体局部剥落，这种情况下，轴承故障特征频率为外环通过频率 f_{bc} 及其倍频，理论计算得到 f_{bc}=218.98Hz。

例程 14-2：利用小波变换检测轴承滚动体故障。

f14_2.m

```
%采样频率
fs=10000;

%轴承滚动体故障信号
fid=fopen('bearingroll.dat','r');%故障
N=1024;
xdata=fread(fid,N,'int16');
fclose(fid);
xdata=(xdata-mean(xdata))/std(xdata,1);

%时域波形
figure(1);
plot(1:N,xdata);
xlabel('样本序号 n/个');
ylabel('电压 V/v');

%db10 小波进行 4 层分解
%一维小波分解
[c,l] = wavedec(xdata,4,'db10');

%重构第 1-4 层细节系数
d4 = wrcoef('d',c,l,'db10',4);
d3 = wrcoef('d',c,l,'db10',3);
d2 = wrcoef('d',c,l,'db10',2);
d1 = wrcoef('d',c,l,'db10',1);

%显示细节信号
figure(2)
subplot(4,1,1);
plot(d4,'LineWidth',2);
ylabel('d4');
subplot(4,1,2);
plot(d3,'LineWidth',2);
ylabel('d3');
subplot(4,1,3);
```

```
plot(d2,'LineWidth',2);
ylabel('d2');
subplot(4,1,4);
plot(d1,'LineWidth',2);
ylabel('d1');
xlabel('样本序号 n/个');

%第 1 层细节信号的包络谱
y=hilbert(d1);
ydata=abs(y);
y=y-mean(y);
nfft=1024;
p=abs(fft(ydata,nfft));
figure(3);
plot((0:nfft/2-1)/nfft*fs,p(1:nfft/2));
xlabel('频率 f/Hz');
ylabel('功率谱 P/W');
```

采集的滚动体故障振动信号时域波形如图 14.8 所示，从波形图上根本看不出是否存在外环故障。

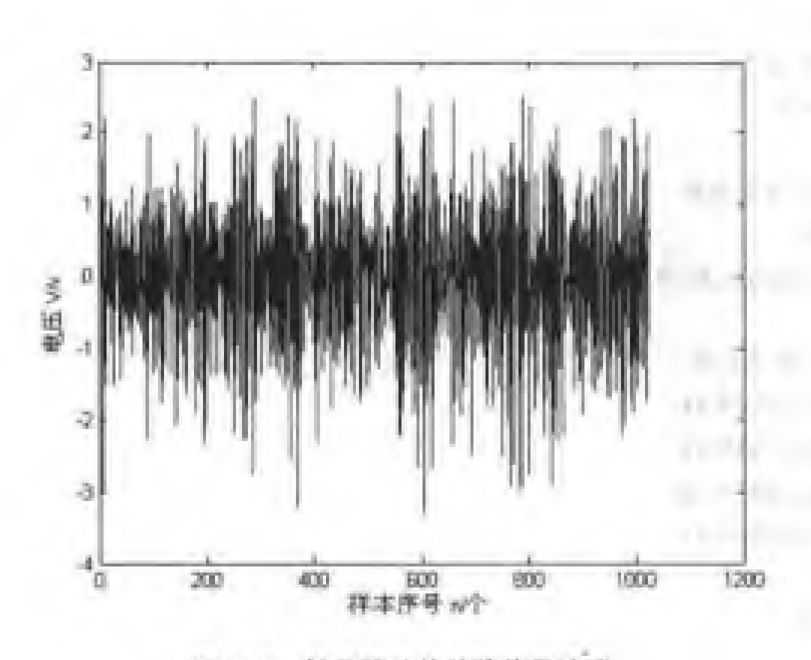


图 14.8 轴承滚动体故障信号波形

对图 14.8 中的信号用 db10 正交小波基进行 4 层小波分解，分解结果如图 14.9 所示，其中 d1～d4 分别表示第一、二、三、四层细节信号。进一步为了提取外环故障特征频率，对第一层细节信号 d1 作 Hilbert 包络并进行谱分析。结果如图 14.10 所示，从功率谱的分析中可以发现频率 215Hz 的存在，通过对照轴承故障特征频率可知，轴承的滚动体发生了故障。

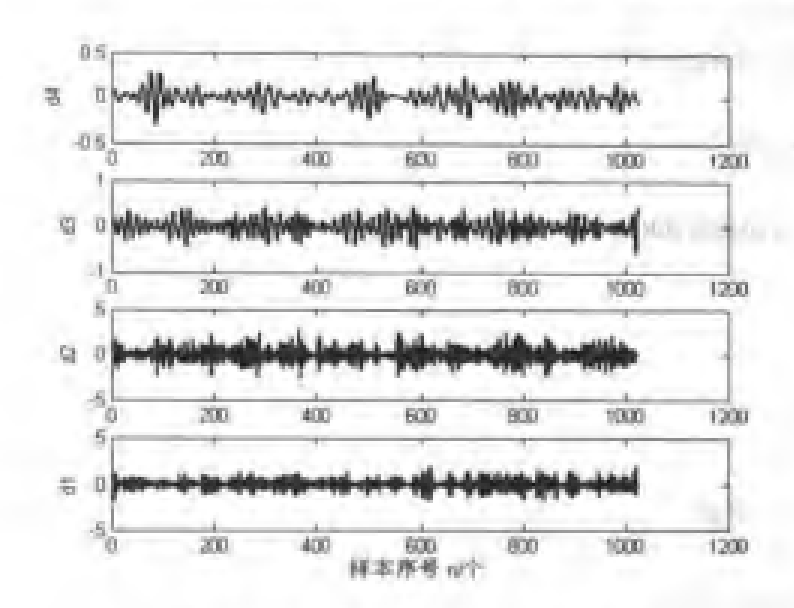


图 14.9　4 层 db10 小波分解细节信号

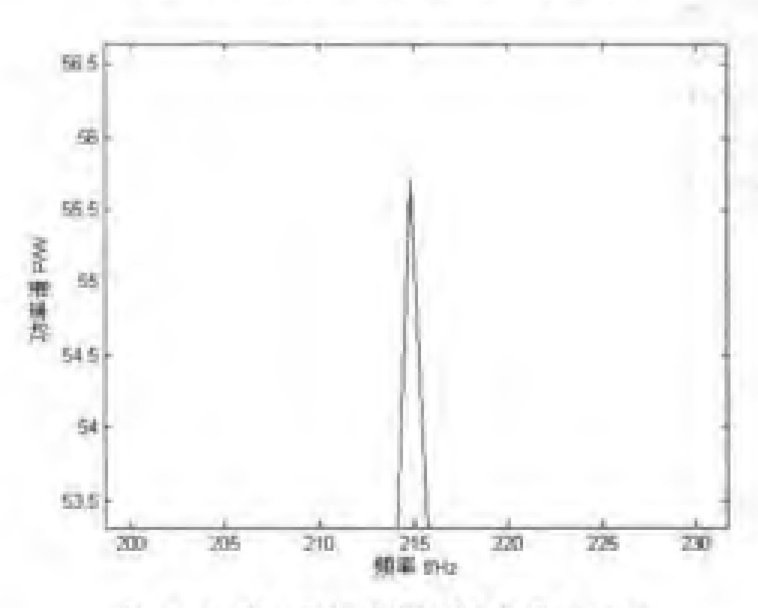


图 14.10　第 1 层细节信号的包络谱（db10）

14.3　小波变换在汽车变速器齿轮故障诊断中的应用

点蚀是齿轮常见的一种故障，在齿轮啮合过程中，齿面上的点蚀会产生突变的冲击脉冲信号，通过检测突变信号可以对确定故障的位置，但是在实测的信号中，突变信号往往受到各种噪声的干扰，利用小波的消噪功能可以检测微弱的故障突变信号。

例程 14-3：利用小波变换检测齿轮点蚀故障。

f14_3.m

```
%采样频率
fs=10000;
```

```
%齿轮点蚀声音信号
fid=fopen('GEAR2.dat','r');%故障
N=1024;
xdata=fread(fid,N,'int16');
fclose(fid);
xdata=(xdata-mean(xdata))/std(xdata,1);
%时域波形
figure(1);
plot(1:N,xdata);
xlabel('时间 t/n');
ylabel('电压 V/v');

%db5 小波进行 4 层分解
%一维小波分解
[c,l] = wavedec(xdata,4,'db5');
a4 = wrcoef('a',c,l,'db5',4);
%重构第 1-4 层细节系数
d4 = wrcoef('d',c,l,'db5',4);
d3 = wrcoef('d',c,l,'db5',3);
d2 = wrcoef('d',c,l,'db5',2);
d1 = wrcoef('d',c,l,'db5',1);

%显示细节号
figure(2)
subplot(5,1,1);
plot(d1,'LineWidth',2);
ylabel('d1');
subplot(5,1,2);
plot(d2,'LineWidth',2);
ylabel('d2');
subplot(5,1,3);
plot(d3,'LineWidth',2);
ylabel('d3');
subplot(5,1,4);
plot(d4,'LineWidth',2);
ylabel('d4');
xlabel('时间 t/s');
subplot(5,1,5);
plot(a4,'LineWidth',2);
ylabel('a4');
xlabel('时间 t/s');
```

采集的齿轮点蚀故障信号时域波形如图 14.11 所示，从图中不能看出突变信号的存在。利用 db5 正交小波基对其进行 4 层分解，结果如图 14.12 所示，从细节信号 d3 中已经明显能

看出周期性突变信号的存在，它对应了点蚀故障引起的周期性冲击信号。

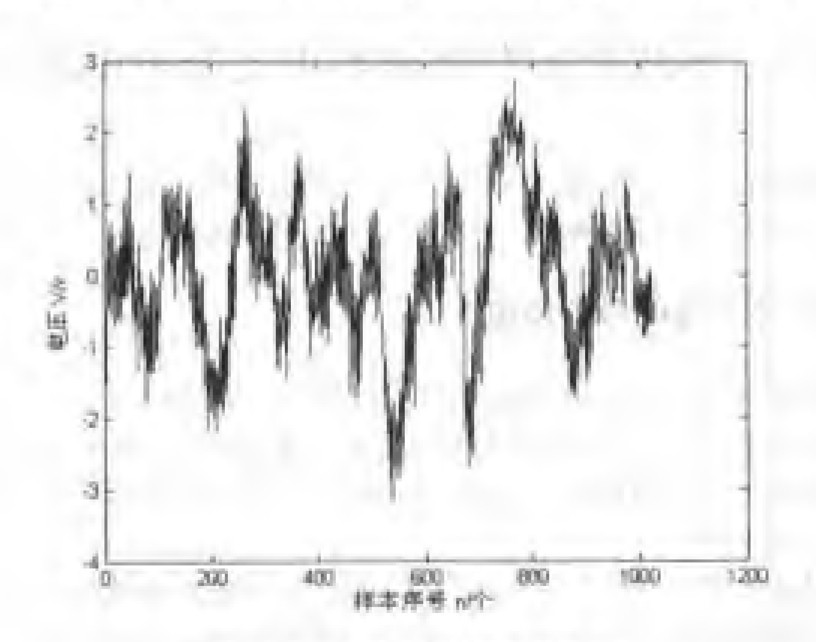


图 14.11　齿轮点蚀故障信号波形

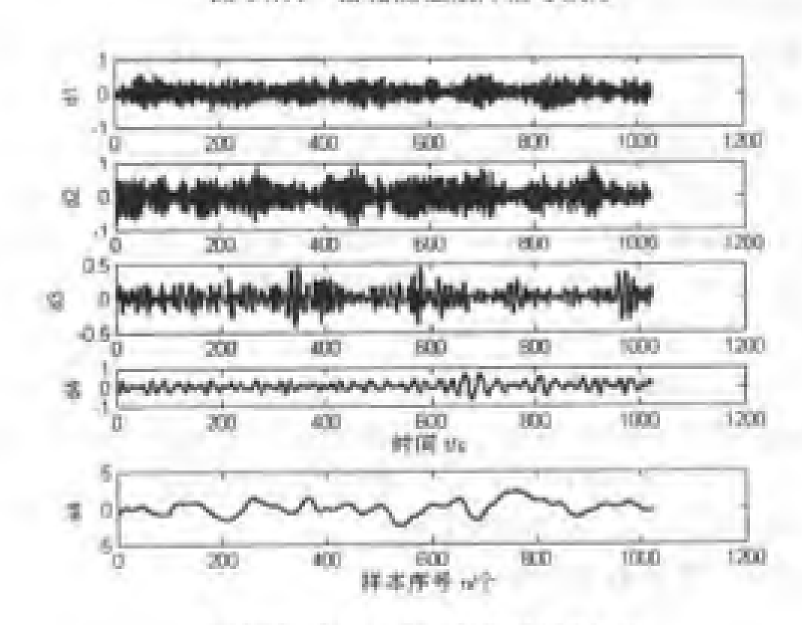


图 14.12　4 层 db5 小波分解细节信号

如何选择小波基函数目前还没有一个理论标准，但是小波变换的小波系数为如何选择小波基函数提供了依据。小波变换后的小波系数表明了小波与被处理信号之间的相似程度，如果小波变换后的小波系数比较大，就表明小波和信号的波形相似程度较大；反之则比较小。另外还要根据信号处理的目的来决定尺度的大小。如果小波变换仅仅要反映信号整体的近似特性，往往选用较大的尺度；反映信号细节上的变换则选用尺度较小的小波。由于小波函数家族成员较多，进行小波变换的目的各异，目前还没有一个通用的标准。不过在实际应用中已取得了一些经验。实际中，Morlet 小波应用领域较广，可以用于信号表示和分类、图像识别和特征提取；墨西哥草帽小波用于系统辨识；样条小波用于材料探伤；Shannon 正交基用于差分方程求解。

14.4 小波变换在路面不平度分析中的应用

汽车是一个复杂的动力学系统，由于不平路面的激励所引起的振动不仅影响汽车的行使平稳性，而且会影响部件的使用寿命，因此对路面不平度的研究是一项重要的工作。

14.4.1 路面不平度的数值模拟

各种道路的路面不平度是引起车辆在运动过程中产生振动的主要原因之一，它是车辆运动分析最基本的输入。为了定量评定车辆振动过程，不仅需要了解车辆动力学系统的性质，而且必须知道道路激励干扰的性质。因此，对路面不平度的描述是进行车辆振动响应计算及悬挂系统优化设计的基础。

时间序列分析是统计学科的一个重要分支内容，在市场价格预测、经济管理、信号处理以及控制工程等领域得到了广泛应用。近些年来，时间序列分析中的自回归模型（AR）开始被用于路面不平度的模拟，取得了良好的效果。

假设 $y(t)$为路面不平度函数，经过采样后得到一维序列$\{y(n)\},n=1,2,\cdots,N$，N 为采样点数，不妨令该序列的均值为零，则其可以用下列的数学模型来表示

$$y(n)=a(1)\times y(n-1)+a(2)\times y(n-2)+\cdots+a(p)\times y(n-p)+\omega(n) \tag{14.4.1}$$

式中ω为一零均值的白噪声，$a(i),i=1,\cdots,p$ 为自回归系数，p 是模型的阶数。

这样，问题就转化为求方程式（14.4.1）的自回归系数 $a(i)$以及模型的阶次 p。目前，求解上述问题的方法有多种，常用的一种方法是根据采集样本 $\{y(n)\}$ 最优选择 $a(i)$，使残差的方差$\Sigma\omega^2(i)$取最小值，优化的方法可以使用最小二乘原理或极大似然原则。

式（14.4.2）是根据实测路面数据拟合的自回归模型，这是一个 AR(7)模型：

$$\begin{aligned}y(n)=&1.0658\times y(n-1)-0.4533\times y(n-2)+0.3940\times y(n-3)-0.1735\times y(n-4)+0.1375\times y(n-5)\\&-0.0551\times y(n-6)+0.0799\times y(n-7)+\omega(n)\end{aligned} \tag{14.4.2}$$

下面利用式（14.4.2）产生路面不平度数据。

例程 14-4：路面不平度的数值仿真。

f14_4.m

```
%零均值的白噪声输入
load wnoise.dat -ascii
N=length(wnoise);
w=wnoise;
x=w;
%路面不平度的模拟
for i=8:N
    x(i)=1.0658*x(i-1)-0.4533*x(i-2)+0.3940*x(i-3)-0.1735*x(i-4)+...
        0.1375*x(i-5)-0.0551*x(i-6)+0.0799*x(i-7)+w(i);
end
```

```
%路面不平度输出
x=(x-mean(x));
%显示白噪声
subplot(2,1,1);
plot(1:N,w);
xlabel('样本序号 n');
ylabel('幅值 A');
%显示路面不平度
subplot(2,1,2);
plot(1:N,x);
xlabel('样本序号 n');
ylabel('幅值 A');
```

模型的零均值白噪声如图 14.13（a）所示。产生的路面不平度仿真结果如图 14.13（b）所示，下一节将利用小波变换对其进行分析，以揭示路面不平度的激励成分。

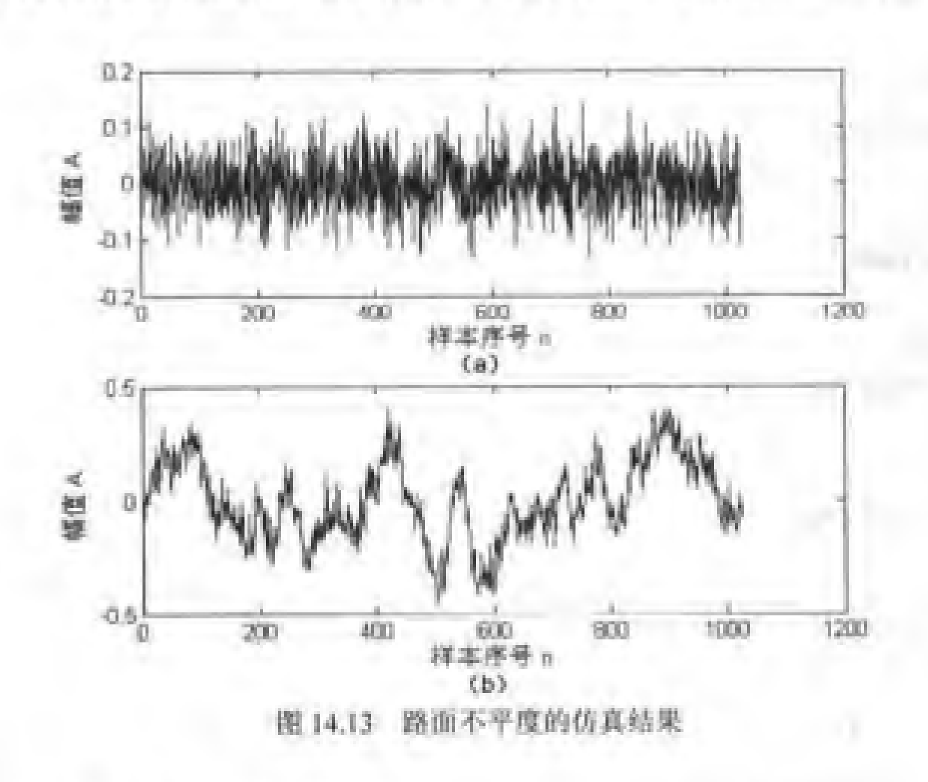


图 14.13 路面不平度的仿真结果

14.4.2 路面不平度的小波分析

路面不平度反映的是车辆对路面的振动响应，所以存在随机性和不稳定性，而且获取的信号中还夹杂着大量的噪声信号。如果直接对采集信号进行分析，必然会造成误差。小波变换具有很好的时频特性和滤波特性，因此利用小波分析来处理路面不平度，不仅可以降低噪声，而且能同时显示信号的时频特性。这样，可以获得对路面不平度更深层次的了解，更能充分揭示车辆的振动状况。

下面应用小波工具箱的函数，利用 db2 小波对路面不平度进行 5 层小波分析。

例程 14-5：路面不平度的小波分析。

f14_5.m

```
%装载数据
load sample.dat -ascii;
xdata=sample;
%d2 小波进行 5 层分解
%一维小波分解
[c,l] = wavedec(xdata,5,'db2');
%重构第 1-4 层细节系数
d5 = wrcoef('d',c,l,'db2',5);
d4 = wrcoef('d',c,l,'db2',4);
d3 = wrcoef('d',c,l,'db2',3);
d2 = wrcoef('d',c,l,'db2',2);
d1 = wrcoef('d',c,l,'db2',1);
%显示细节号
figure(2)
subplot(5,1,1);
plot(d1,'LineWidth',2);
ylabel('d1');
subplot(5,1,2);
plot(d2,'LineWidth',2);
ylabel('d2');
subplot(5,1,3);
plot(d3,'LineWidth',2);
ylabel('d3');
subplot(5,1,4);
plot(d4,'LineWidth',2);
ylabel('d4');
subplot(5,1,5);
plot(d5,'LineWidth',2);
ylabel('d5');
xlabel('样本序号 n');
```

利用 db2 小波对路面不平度进行 5 层小波分解，得到的 1～5 层的细节信号如图 14.14 所示。从图可以看出，第 1 层细节信号的幅值较小，第 2 层细节信号的幅值有所增加，而第 3～5 层细节信号不但最大幅值有所增加，而且幅值的包络线构成了低频成分。特别地，第 5 层细节信号可以近似为正弦或余弦信号。

下面对分解后的各层小波进行谱分析，以揭示其谱特性。

例程 14-6：各层小波的谱分析。

```
>> WaveletPSD(d1)
>> WaveletPSD(d2)
>> WaveletPSD(d3)
```

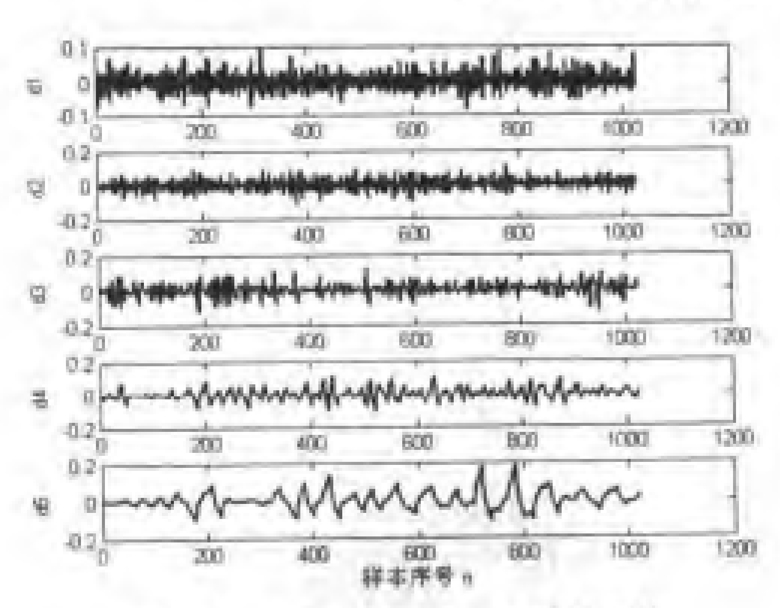


图 14.14　路面不平度的 5 层 db2 小波分析

```
>> WaveletPSD(d4)
>> WaveletPSD(d5)
```

WaveletPSD.m

```
function WaveletPSD(x)
%输入 x-各层的小波
nfft=1024;
fs=1;
y=abs(fft(x,nfft));
plot((0:nfft/2-1)/nfft*fs,y(1:nfft/2));
xlabel('频率 f/Hz');
ylabel('功率谱 P/W');
```

计算得到第 1 层小波的功率谱如图 14.15 所示，可以看出，路面不平度的高频成分是一

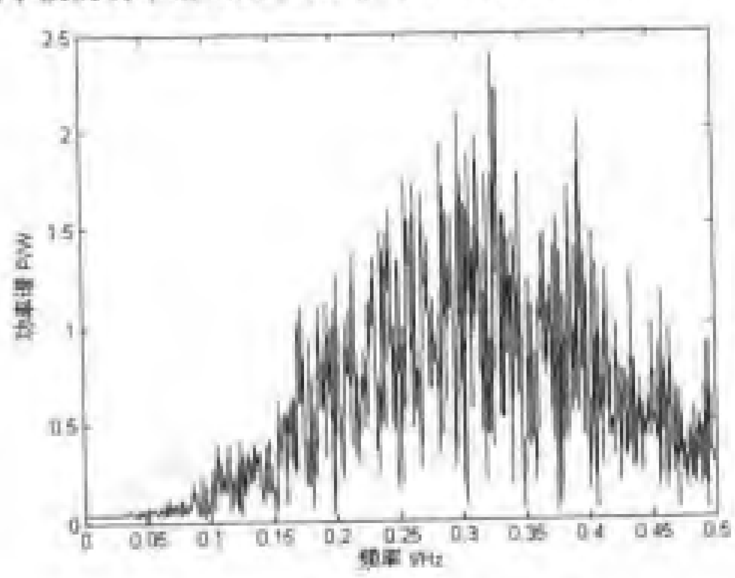


图 14.15　第一层小波的功率谱

宽带过程，其频谱近似于一白噪声谱。第 2 层小波信号的功率谱如图 14.16 所示。由图可见，第 2 层小波的频谱也近似为一宽带过程，但它与第一层小波相比，具有一定的明显峰值。

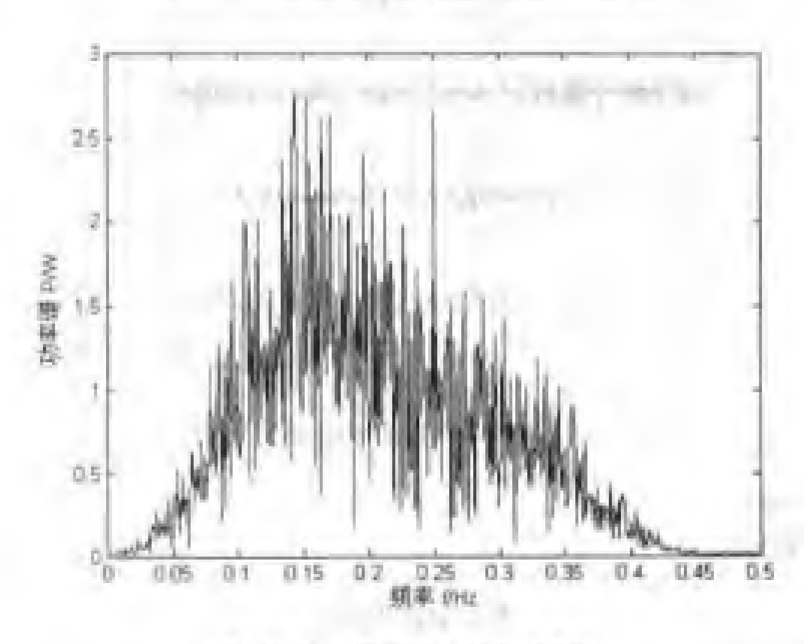


图 14.16 第二层小波的功率谱

第 3~5 层小波的功率谱分别如图 14.17、图 14.18 和图 14.19 所示。由图可见，低频段的频谱为一个窄带谱，这说明路面不平度的低频成分是一个窄带过程，而且还可以看见，功率谱中还有具有一定的高频成分，只不过其幅值比较小。特别是第 5 层小波，它的功率谱中的低频段的频谱近似于一个理想的窄带谱，而且峰值比较大，这与前面假设其为一正弦或余弦信号是一致的。

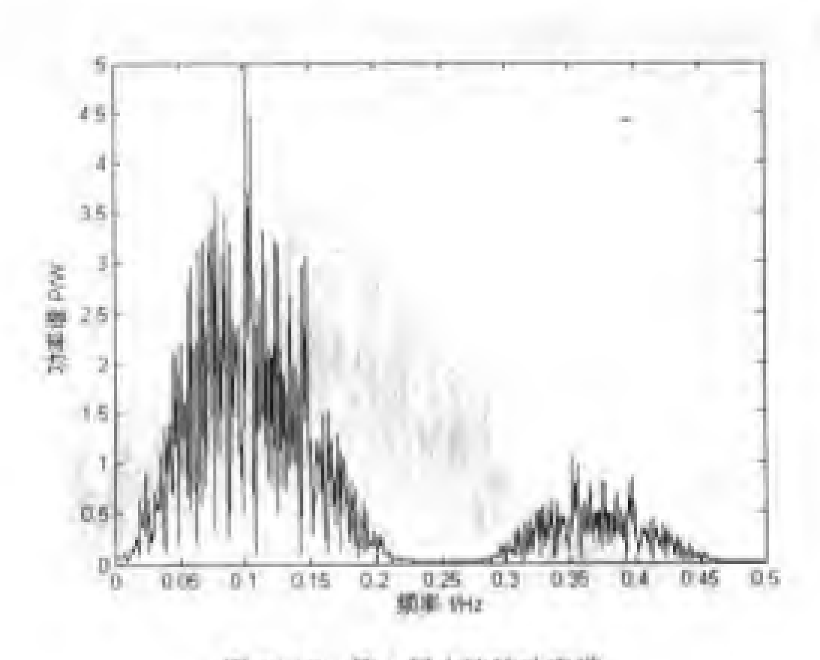


图 14.17 第 3 层小波的功率谱

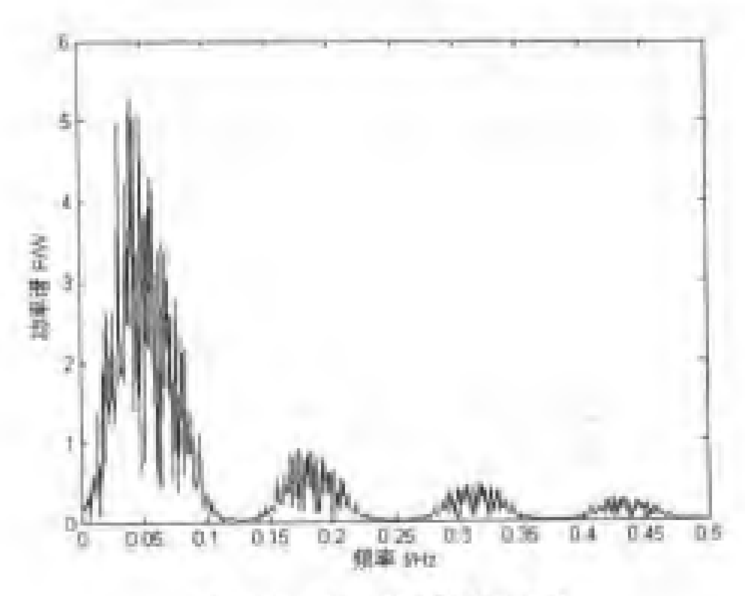


图 14.18　第 4 层小波的功率谱

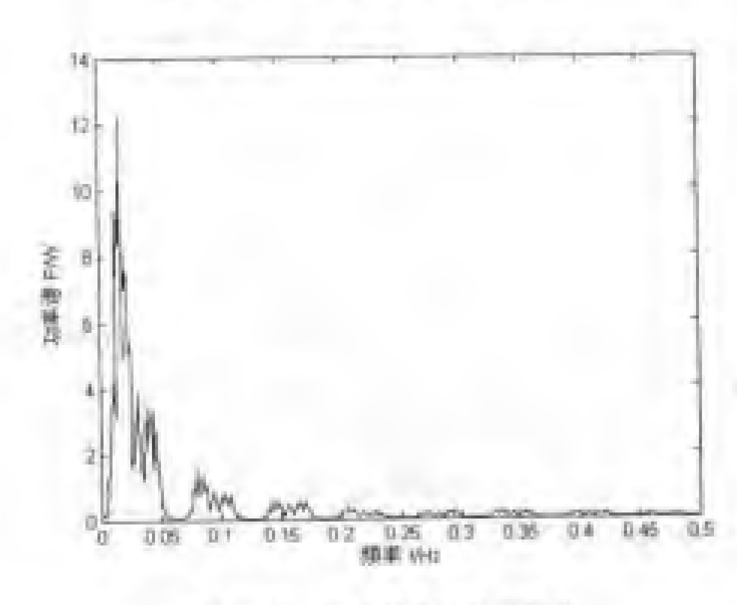


图 14.19　第 5 层小波的功率谱

14.5　小波变换在车辆非平稳振动分析中的应用

为了改善车辆的行使平稳性，人们一直对车辆系统的振动进行研究。实际中，车辆大多数时间是在变车速条件下行使的，比如起步加速和制动停止等过程，此时系统的振动是非平稳的。小波变换是处理非平稳信号的有效手段，因此利用小波变换可以分析系统的各级振动信号。

14.5.1　四轮路面输入的数值仿真

四轮路面模型的表达式为

$$\dot{\mathrm{I}}(t)=\mathrm{A}\bullet\mathrm{I}(t)+2\pi\sqrt{G_0 v(t)}\mathrm{B}_1\bullet\mathrm{x}+2\pi\sqrt{G_0 v(t)}\mathrm{B}_2\bullet w(t)$$
$$\dot{\mathrm{x}}=\mathrm{C}\bullet\mathrm{x}+\mathrm{D}\bullet w(t) \tag{14.5.1}$$

式中 I 为 4 个路面输入位移组成的向量，x 为状态转移向量，$w(t)$为零均值的白噪声，A、B、C、D 均为系数矩阵。

各系数矩阵定义如下：

$$\mathrm{A}=\begin{bmatrix}-a\bullet v(t) & 0 & 0 & 0\\ 0 & -a\bullet v(t) & 0 & 0\\ 0 & 0 & -a\bullet v(t) & 0\\ 0 & 0 & 0 & -a\bullet v(t)\end{bmatrix},\quad \mathrm{B}_1=\begin{bmatrix}0 & 0 & 0 & 0\\ a_1-a_2\dfrac{b_1}{b_2} & a_0-a_2\dfrac{b_0}{b_2} & 0 & 0\\ 0 & 0 & 1-\dfrac{T(t)}{2} & 0\\ a_1-a_2\dfrac{b_1}{b_2} & a_0-a_2\dfrac{b_0}{b_2} & 1-\dfrac{T(t)}{2} & 0\end{bmatrix}$$

$$\mathrm{B}_2=\begin{bmatrix}1\\ a_2/b_2\\ 1\\ a_2/b_2\end{bmatrix},\quad \mathrm{C}=\begin{bmatrix}-b_1/b_2 & -b_0/b_2 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 0 & -6/T(t) & -12/T^2(t)\\ 0 & 0 & 1 & 0\end{bmatrix},\quad \mathrm{D}=\begin{bmatrix}1/b_2\\ 0\\ 12/T^2(t)\\ 0\end{bmatrix}$$

$$\mathrm{x}=\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix},\mathrm{I}(t)=\begin{bmatrix}I_1(t)\\ I_2(t)\\ I_3(t)\\ I_4(t)\end{bmatrix} \tag{14.5.2}$$

式中 a,b 为道路系数，且 $a=2\pi n_0, b=2\pi\sqrt{G_0}$，$v(t)$为瞬时车速，$T(t)=L/v(t)$，$n_0$ 为空间截断频率 n_0=0.01～0.1c/m，G_0 为路面不平度系数，表示不同的道路等级。

下面以加速和制动两个典型工况为例来研究车辆的非平稳振动过程。假设车辆从初始速度 v_0=0 开始，以加速度 $a_1^*=2.5\text{m/s}^2$ 匀加速到 v_m=60km/h，然后再以 $a_2^*=-4.5\text{m/s}^2$ 匀减速到 v_f=0，则此过程中任意 t 时刻的瞬时车速为：

$$v(t)=\begin{cases}v_0+a_1^*t & 0\le t\le v_m/a_1^*\\ v_m+a_2^*(t-v_m/a_1^*) & v_m/a_1^*\le t\le v_m/a_1^*-v_m/a_2^*\end{cases} \tag{14.5.3}$$

下面利用龙格-库塔法求解式（14.5.1），得到四轮路面时间序列。

方程式中的参数设置如下：

$$a_1^*=2.5, a_2^*=-4.5, G_0=2.56\times10^{-4}, n_0=0.05, L=20$$
$$a=2\pi n_0, a_0=3.1815, a_1=0.2063, a_2=0.0108$$
$$b=2\pi\sqrt{G_0}, b_0=3.223, b_2=0.59$$

例程 14-7：利用龙格-库塔法求解四轮路面时间序列。

f14_6.m

```
dt=0.005;
N=2000;
global w;
```

```
load w1.dat -ascii;
w=w1;
%初始值
t0=0.00001;
y0=[0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001]';

[t,y]=LK2('LK1',t0,dt,N-1,y0);

%绘时域图
figure(1);
subplot(2,2,1);
y1=y(1,:);
ts=0:N-1;
plot(dt*ts,y1);
xlabel('时间    t / s');
ylabel('路面位移输入 I1 / m');
subplot(2,2,2);
y2=y(2,:);
plot(dt*ts,y2);
xlabel('时间    t / s');
ylabel('路面位移输入 I2 / m');
subplot(2,2,3);
y3=y(3,:);
plot(dt*ts,y3);
xlabel('时间    t / s');
ylabel('路面位移输入 I3 / m');
subplot(2,2,4);
y4=y(4,:);
plot(dt*ts,y4);
xlabel('时间    t / s');
ylabel('路面位移输入 I4 / m');
```

相关函数为：

LK1.m：状态方程式

LK2.m：龙格-库塔法

v.m：瞬时速度

LK1.m

```
function ydot=LK1(t,n,y)
global w;
%参数值
a0=3.1815;
a1=0.2063;
```

```
a2=0.0108;
b0=3.223;
b1=0.59;
b2=0.0327;
G0=2.56*10^(-4);
n0=0.05;
L=20;
a=2*pi*n0;
b=2*pi*sqrt(G0);
temp=2*pi*sqrt(G0*v(t));
%状态方程组
ydot(1)=-a*v(t)*y(1)+temp*w(n);
ydot(2)=-a*v(t)*y(2)+temp*(a1-a2*b1/b2)*y(5)+temp*(a0-a2*b0/b2)*y(6)+temp*a2/b2*w(n);
ydot(3)=-a*v(t)*y(3)+temp*(1-L/(2*v(t)))*y(7)+temp*w(n);
ydot(4)=-a*v(t)*y(4)+temp*(a1-a2*b1/b2)*y(5)+temp*(a0-a2*b0/b2)*y(6)+...
     temp*(1-L/(2*v(t)))*y(7)+temp*a2/b2*w(n);
ydot(5)=-b1/b2*y(5)-b0/b2*y(6)+w(n)/b2;
ydot(6)=y(5);
ydot(7)=-6*v(t)/L*y(7)-12*(v(t))^2/L^2*y(8)+12*(v(t))^2/L^2*w(n);
ydot(8)=y(7);
ydot=[ydot(1) ydot(2) ydot(3) ydot(4) ydot(5) ydot(6) ydot(7) ydot(8)];
%输出
ydot=ydot';
```

LK2.m

```
function [tout,yout]=LK2(funname,t0,step,n,y0)
if nargin~=5
   error('wrong number of parameters in using pend2(funname,t0,step,n,y0)');
end;

y=y0; tout(1)=t0; yout(:,1)=y0;
for i=1:n
   ttemp1=tout(i);
   ttemp2=ttemp1+step/2;
   k1=feval(funname,ttemp1,i,y);
   k2=feval(funname,ttemp2,i,y+(step/2)*k1);
   k3=feval(funname,ttemp2,i,y+(step/2)*k2);
   k4=feval(funname,ttemp1+step,i+1,y+step*k3);

   y=y+(step/6)*(k1+2*k2+2*k3+k4);
   yout(:,i+1)=y;
   tout(i+1)=ttemp1+step;
end
```

```
tout=tout';
```

v.m

```
function y=v(t)
a1=2.5;
a2=-4.5;
v0=0;
vm=60;
if ((t>=0)&(t<vm/a1))
    y=v0+a1*t;
else if ((t>=vm/a1)&(t<(vm/a1-vm/a2)))
        y=vm+a2*(t-vm/a1);
    else
        disp('t is out of range!');
    end
end
```

利用龙格-库塔方法求解方程式（14.5.1），得到四个轮子的路面不平度时间序列，如图 14.20 所示。

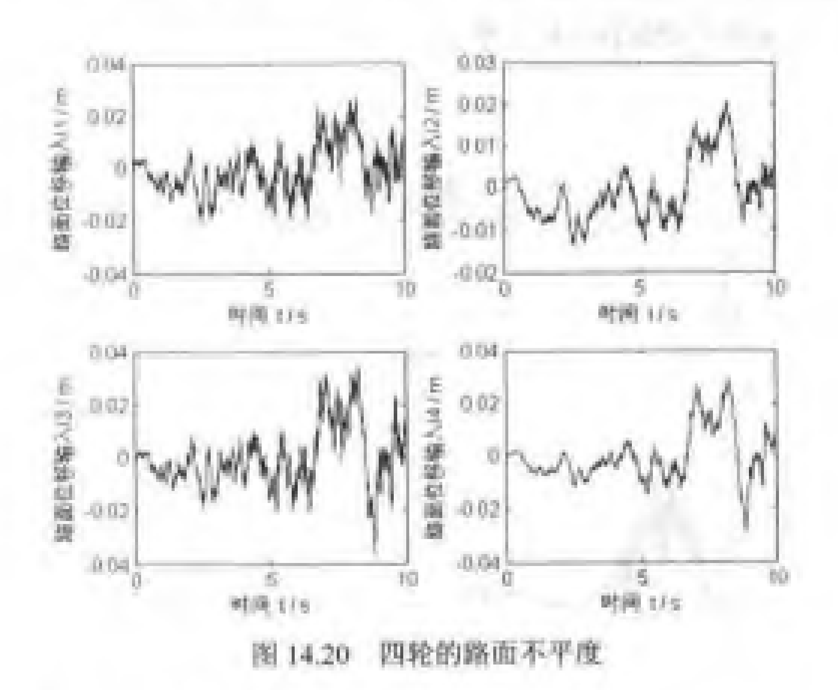


图 14.20　四轮的路面不平度

14.5.2　车辆振动的小波分析

根据文献[30]，整车的力学模型可以用一个 8 自由度的模型来表示，其表达式如下：

$$M\ddot{z} + C\dot{z} + Kz = P \cdot l(t) \tag{14.5.4}$$

式中 M 为系统质量矩阵；C 为系统阻尼系数矩阵；K 系统刚度矩阵；P 为路面位移和激励的转换矩阵；z 为系统响应向量（8 个变量，定义如表 14.1 所示）。

表 14.1　　8 个自由度的定义

序　号	物理符号	物 理 意 义
z_1	Z_B	簧载质量质心的垂直位移
z_2	θ_B	簧载质量绕质心的横向角位移
z_3	ϕ_B	簧载质量绕质心的纵向角位移
z_4	Z_f	前非簧载质量质心的垂直位移
z_5	Z_r	后非簧载质量质心的垂直位移
z_6	θ_f	前非簧载质量质心的横向角位移
z_7	θ_r	后非簧载质量质心的纵向角位移
z_8	Z_s	座椅系统质心的垂直位移

令 $y=[z,\dot{z}]^T$，将式（14.5.4）写成状态方程的形式，如下所示

$$\dot{y}_{16\times1}=\begin{bmatrix}0_{8\times8} & E_{8\times8}\\ B_{8\times8} & A_{8\times8}\end{bmatrix}\cdot y+\begin{bmatrix}0_{8\times1}\\ \Gamma'(t)_{8\times1}\end{bmatrix} \tag{14.5.5}$$

式中 $\Gamma'(t)=P\cdot I(t)$。

同样利用龙格-库塔方法可以求解方程式（14.5.5），得到整车的非平稳振动响应。其中座椅质心的响应信号及其功率谱如图 14.21 所示。

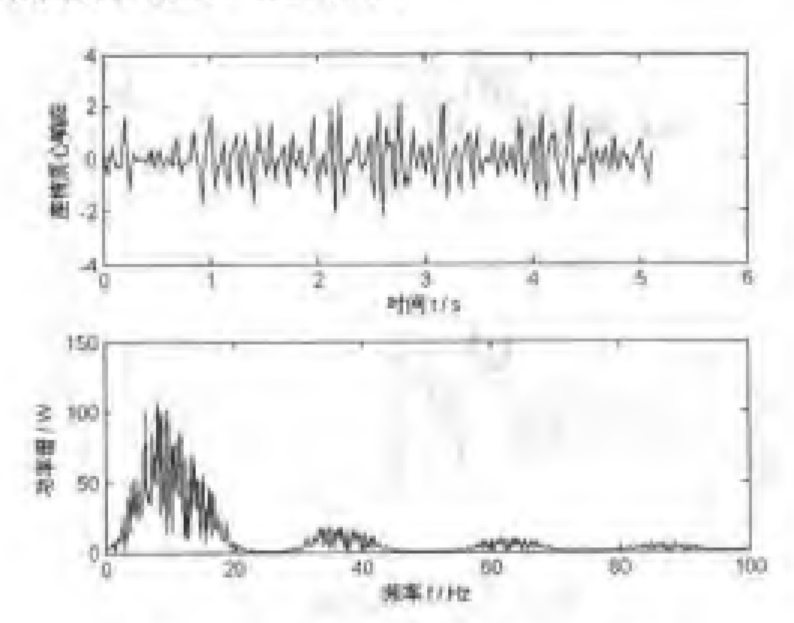


图 14.21　座椅质心响应信号的波形及其功率谱

进一步，对座椅质心响应信号进行四层小波分解。

例程 14-8：座椅质心响应信号的小波分析。

f14_7.m

```
%装载数据
load seat.dat -ascii
```

```
dt=0.005;
N=length(seat);
subplot(2,1,1);
plot((1:N)*dt,seat);
xlabel('时间  t / s');
ylabel('座椅质心响应');

nfft=1024;
y=abs(fft(seat,nfft));
fs=1/dt;
subplot(2,1,2);
plot((0:nfft/2-1)/nfft*fs,y(1:nfft/2));
xlabel('频率  f / Hz');
ylabel('功率谱  / W');

%d5 小波进行 4 层分解
%一维小波分解
[c,l] = wavedec(seat,4,'db5');
%重构第 1-4 层细节系数
d4 = wrcoef('d',c,l,'db5',4);
d3 = wrcoef('d',c,l,'db5',3);
d2 = wrcoef('d',c,l,'db5',2);
d1 = wrcoef('d',c,l,'db5',1);

%显示细节号
figure(2);
subplot(4,1,1);
plot(d1,'LineWidth',2);
ylabel('d1');
subplot(4,1,2);
plot(d2,'LineWidth',2);
ylabel('d2');
subplot(4,1,3);
plot(d3,'LineWidth',2);
ylabel('d3');
subplot(4,1,4);
plot(d4,'LineWidth',2);
ylabel('d4');
xlabel('样本序号  n');
```

利用 db5 小波对座椅质心响应信号进行四层小波分解，得到的细节信号如图 14.22 所示。由图 14.21 可见，原始信号的振动能量主要集中在 6.25～12.5Hz 的频段范围内，进一步计算各层小波的功率谱如图 14.23 所示，可见各级分解信号功率谱的主要峰值位于不同的频段内。

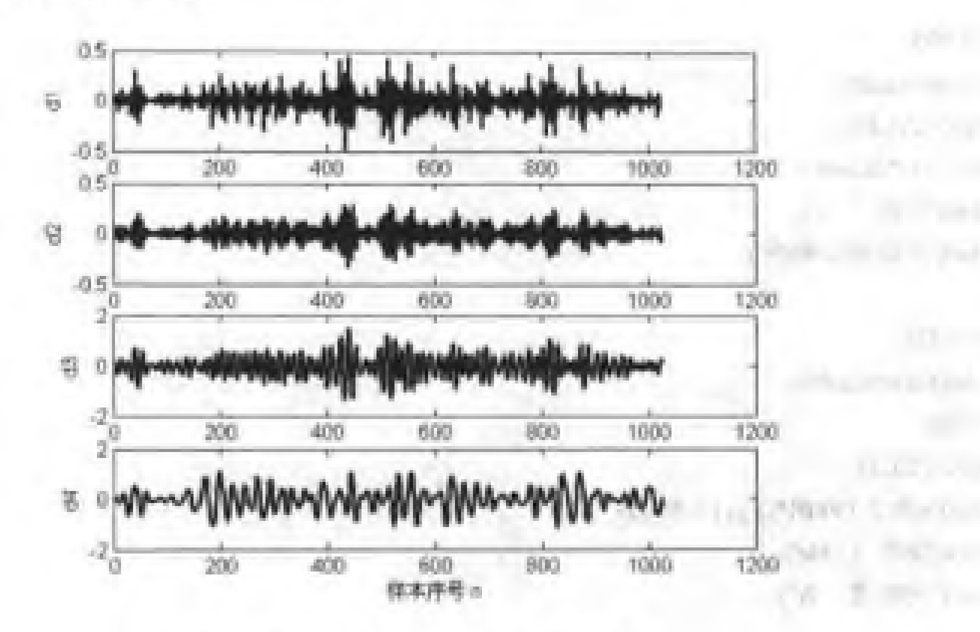


图 14.22　四层小波分解后的细节信号

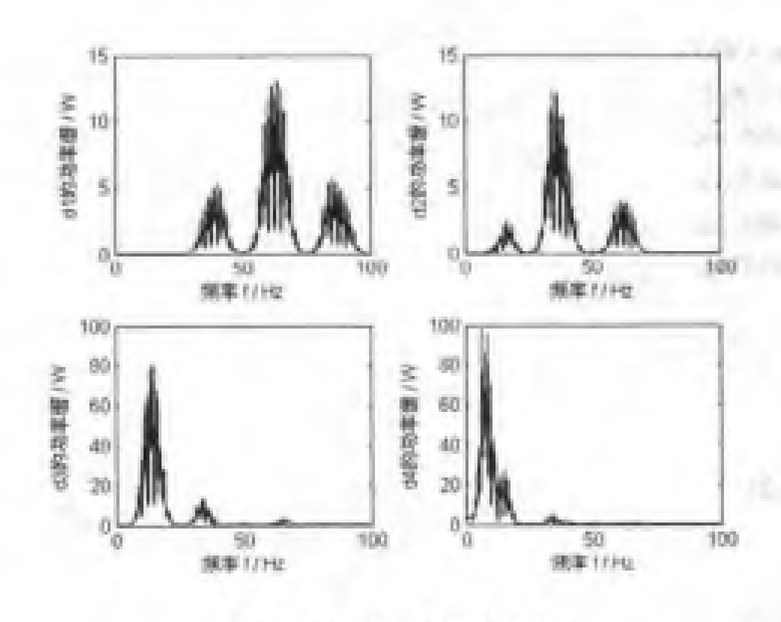


图 14.23　各层小波的功率谱

第 15 章　Matlab 提升小波变换与应用初步

传统的基于卷积的离散小波变换缺点是计算量大，计算复杂度高，对存储空间的要求高，不利于硬件实现。提升小波的出现有效地解决了这一问题。提升算法相对于 Mallat 算法而言，是一种更为快速有效的小波变换实现方法，被誉为第二代小波变换。它不依赖于傅立叶变换，继承了第一代小波的多分辨率特征，小波变换后的系数是整数，计算速度快，计算时无需额外的存储开销，Daubechies 已经证明，任何离散小波变换或具有有限长滤波器的两阶滤波变换都可以被分解成为一系列简单的提升步骤，所有能够用 Mallat 算法实现的小波，都可以用提升算法来实现。提升算法给出了双正交小波简单而有效的构造方法，使用了基本的多项式插补来获取信号的高频分量，之后通过构建尺度函数来获取信号的低频分量。"提升"算法的基本思想是，将现有的小波滤波器分解成基本的构造模块，分步骤完成小波变换。

15.1　提升小波变换的基本原理

15.1.1　提升算法

基于提升算法的小波变换使我们能够用一种简单的方法去解释小波的基本理论，而第一代小波变换都可找到等效的提升方案。提升方案把第一代小波变换过程分为以下 3 个阶段：分解（Split）、预测（Predict）和更新（Update）。提升算法的分解和重构，如图 15.1 所示。

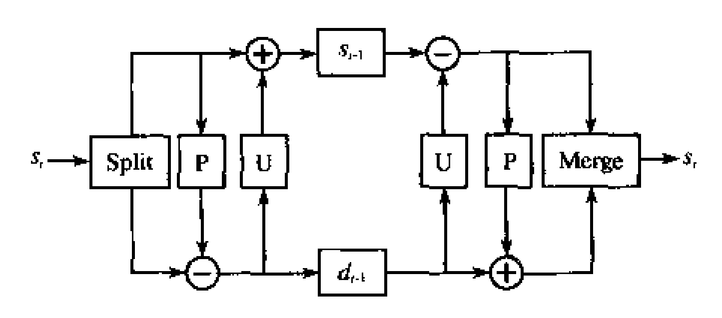


图 15.1　提升算法的分解和重构

（1）分解。将输入信号 s_i 分为 2 个较小的子集 s_{i-1} 和 d_{i-1}，d_{i-1} 也称为小波子集。最简单的分解方法是将输入信号 s_i 根据奇偶性分为 2 组，这种分裂所产生的小波称为懒小波（Lazy Wavelet），分解过程表示为 $F(s_i)=(s_{i-1},d_{i-1})$，其中 $F(s_i)$ 为分解过程。

（2）预测。在基于原始数据相关性的基础上，用偶数序列 s_{i-1} 的预测值 $P(s_{i-1})$ 去预测（或者内插）奇数序列 d_{i-1}，即将滤波器 P 对偶数信号作用以后作为奇信号的预测值，奇信号的实际值与预测值相减得到残差信号。实际中，虽然不可能从子集 s_{i-1} 中准确地预测子集 d_{i-1}，但是 $P(s_{i-1})$

有可能很接近 d_{i-1}，因此可以使用 $P(s_{i-1})$和 d_{i-1} 的差来代替原来的 d_{i-1}，这样产生的 d_{i-1} 比原来的 d_{i-1} 包含更少的信息，于是得到 $d_{i-1}=d_{i-1}-P(s_{i-1})$。这里，已经可以用更小的子集 s_{i-1} 和小波子集 d_{i-1} 来代替原信号集 s_i。重复分解和预测过程，经 n 步以后原信号集可用$\{s_n,d_n,\cdots,s_1,d_1\}$来表示。

（3）更新。为了使原信号集的某些全局特性在其子集 s_{i-1} 中继续保持，必须进行更新。更新的思想是要找一个更好的子集 s_{i-1}，使得它保持原图的某一标量特性 $Q(x)$（例如均值、消失矩等不变），即有 $Q(s_{i-1})=Q(s_i)$。可以利用已经计算的小波子集 d_{i-1} 对 s_{i-1} 进行更新，从而使得后者保持特性 $Q(x)$，即要构造一个算子 U去更新 s_{i-1}。定义如下：

$s_{i-1}=s_{i-1}+U(d_{i-1})$。

从上述可以知道，提升方法可以实现原位运算，即该算法不需要除了前级提升步骤的输出之外的数据，这样在每个点都可以用新的数据流替换旧的数据流。当重复使用原位提升滤波器组时，就获得了交织的小波变换系数。

15.1.2 提升框架下的整数小波变换

提升方法是由 Wim Sweldens 在 1996 年首先提出和发展起来的。最初提升方法是用来对一个给定离散小波变换进行改善以得到一些特殊的性质。后来采用提升方法，就可以用简单的方法来计算任意小波变换，即任意小波变换都可以分解成简单的提升步骤。由于数字信号一般是整数数据序列，但小波变换后的结果为浮点数据，由于量化精度的限制，在图像处理中一般需要将浮点数据转化为整数数据，从而导致量化误差。因此，要提高变换的准确性，保证变换是整数—整数变换（integer-integer transform）就极为重要。而提升方法正是针对整数—整数变换的，同时也是计算逆小波变换（inverse wavelet transform）的一种有效方法。基于提升格式的第二代小波有如下特点：

① 本位操作：所有运算可作本位操作，节省内存；

② 效率高：利用复合复制，减少了浮点运算量；

③ 并行性：一个上升步骤中的所有操作是并行的，而多个上升步骤之间是串行的；

④ 逆变换：逆变换只须简单地改变代码执行的先后顺序，具有与正向变换相同的计算复杂性；

⑤ 通用性：由于变换过程中不必依赖 Fourier 分析，很容易推广到一般性应用领域；

⑥ 易于构造非线性小波变换（如整数变换）。

提升框架的实质是对经典小波滤波器采用 Euclidean 算法的多相矩阵分解。式（15-1）与式（15-2）给出了提升框架下小波分析滤波器的多相矩阵表示：

$$p(z)=\prod_{i=1}^{m}\begin{bmatrix}1 & s_i(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ t_i(z) & 1\end{bmatrix}\begin{bmatrix}K & 0\\ 0 & 1/K\end{bmatrix} \tag{15-1}$$

$$\tilde{p}(z)=\prod_{i=1}^{m}\begin{bmatrix}1 & 0\\ -s_i(z^{-1}) & 1\end{bmatrix}\begin{bmatrix}1 & -t_i(z^{-1})\\ 0 & 1\end{bmatrix}\begin{bmatrix}1/K & 0\\ 0 & K\end{bmatrix} \tag{15-2}$$

其中 $p(z)$与 $\tilde{p}(z)$ 被定义为分析滤波器，$s_i(z)$与 $t_i(z)$代表 Laurent 多项式。K 为 1，即缩放因子。基于上述多相矩阵分解，提升框架通过分裂、预测和更新 3 个步骤将原始信号 s_j 小波分解成为低频信号 s_{j-1} 与高频细节信号 d_{j-1}。如图 15.2 所示，整数小波变换的实质是对式（15-1）与式（15-2）给出的 Laurent 多项式 $s_i(z)$与 $t_i(z)$或$-s_i(z^{-1})$与$-t_i(z^{-1})$分别做尾数的截断取整，这里用$\lfloor\cdot\rfloor$表示 IWT 中对 Laurent 多项式的取整操作。图 15.3 给出了几种基于提升框架的 IWT

前向变换实施方法。其各参量定义如下：x/y 表明 IWT 的低通分析滤波器长度为 x，高通分析滤波器长度为 y。$d_0(n)$与 $S_0(n)$分别是待分解信号的奇序列与偶序列。$d(n)$与 $s(n)$分别代表一次前向变换后信号的奇序列与偶序列，K 称为缩放因子或缩放常数。

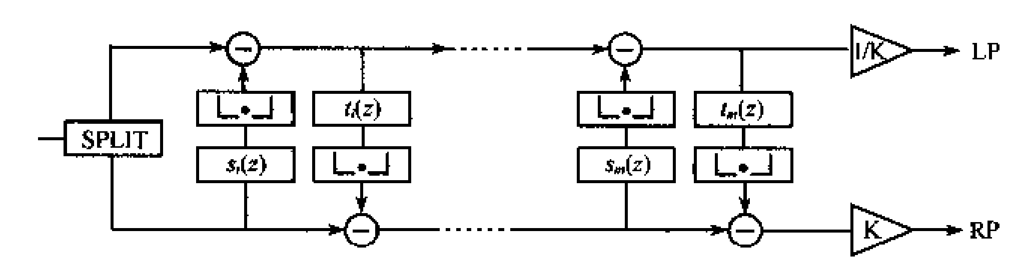


图 15.2 基于提升框架的 IWT 前向变换方框图

名称 (x/y)	IWT 前向变换	说明
5/3	$\begin{cases} d[n] = d_0[n] - \lfloor 1/2(s_0[n+1] + s_0[n]) \rfloor \\ s[n] = s_0[n] + \lfloor 1/4(d[n] + d[n-1]) + 1/2 \rfloor \end{cases}$	K=1
13/7	$\begin{cases} d[n] = d_0[n] + \lfloor 1/16((s_0[n+2] + s_0[n-1]) - 9(s_0[n+1] + s_0[n]) + 1/2) \rfloor \\ s[n] = s_0[n] + \lfloor 1/32(9(d[n] + d[n-1]) - (d[n+1] + d[n-2]) + 1/2 \rfloor \end{cases}$	K=1
6/14	$\begin{cases} d_0[n] = d_0[n] - s_0[n] \\ s[n] = s_0[n] + \lfloor 1/16(8d_1[n] + d_1[n-1] - d_1[n+1]) + 1/2 \rfloor \\ d[n] = d_1[n] + \lfloor 1/16(s[n+2] - s[n-2] + 6(s[n-1] - s[n+1]) + 1/2 \rfloor \end{cases}$	K=1
Db9/7	$d_1[n] = d_0[n] + \lfloor \alpha(s_0[n+1] + s_0[n]) + 1/2 \rfloor$ $s_1[n] = s_0[n] + \lfloor \beta(d_1[n] + d_1[n-1]) + 1/2 \rfloor$ $d[n] = d_1[n] + \lfloor \gamma(s_1[n+1] + s_1[n]) + 1/2 \rfloor$ $s[n] = s_1[n] + \lfloor \delta(d_0[n-1] + d_0[n]) + 1/2 \rfloor$	$\alpha = -1.586134342$ $\beta = -0.05298011854$ $\gamma = 0.8829110762$ $\delta = 1.149604398$ $K = 1.149604398$

图 15.3 几种 IWT 的前向变换

15.2 Matlab 提升小波变换函数

可以使用提升方法来设计新小波，这些提升方法允许“整数到整数”小波变换，并使用不同长度的低通和高通分解滤波器进行小波变换。小波工具箱 3.0 版本中包括的新型提升函数主要有 5 组，如表 15.1 所示。

表 15.1 提升小波变换函数

函 数 名	函 数 名	函 数 意 义
提升方案函数	addlift	向添加原始或双重提升步骤
	displs	显示提升方案
	lsinfo	提升方案信息

续表

函 数 名	函 数 名	函 数 意 义
双正交四联滤波器	bswfun	双正交尺度和小波函数
	filt2ls	将四联滤波器变换为提升方案
	liftfilt	在四联滤波器.上应用基本提升方案
	ls2filt	将提升方案变换为四联滤波器
正交或双正交小波及懒小波	liftwave	提供常用小波的提升方案
	wave2lp	提供小波的劳伦多项式
	wavenames	提供用于 LWT 的小波名信息
提升小波变换和反变换	lwt	一维提升小波变换
	lwt2	二维提升小波变换
	lwtcoef	提取或重构一维 LWT 小波系数
	lwtcoef2	提取或重构二维 LWT 小波系数
	ilwt	一维提升小波反变换
	ilwt2	二维提升小波反变换
劳伦多项式和矩阵	laurmat	劳伦矩阵类 LM 的构造器
	laurpoly	劳伦多项式类 LP 的构造器

1. addlift 函数

【函数功能】添加原始或双重提升步骤。

【语法格式】

① LSN = addlift(LS,ELS)

② LSN = addlift(LS,ELS,'begin')

③ LSN = addlift(LS,ELS,'end')

【使用说明】格式①返回新的提升方案 LSN，它通过增加基本提升步骤 ELS 到提升方案 LS。

格式②预先考虑指定的基本提升步骤。

ELS 是单元阵列：{TYPEVAL, COEFS, MAX_DEG}或者结构：

struct('type',TYPEVAL,'value',LPVAL)。其中 LPVAL = laurpoly(COEFS, MAX_DE)。

LSN = addlift(LS,ELS,'end') 等同于 addfilt(LS,ELS)。

如果 ELS 是一串基本提升步骤，被存储在单元阵列或者结构阵列中，然后这些基本提升步骤被增加到 LS。

2. displs 函数

【函数功能】显示提升方案。

【语法格式】S = displs(LS,FRM)

【使用说明】该函数返回描述提升方案 LS 的字符串。格式字符串 FRM 产生 S。displs(LS) I 等同于 DISPLS(LS,'%12.8f')。

3. lsinfo 函数

【函数功能】提升方案信息。

【语法格式】lsinfo

【使用说明】lsinfo 显示以下的提升方案的信息。提升方案 LS 是 $N\times3$ 单元阵列，前面的 $N-1$ 行是基本的提升步骤（ELS）。最后一行是 LS 的归一化。

每个 ELS 具有的格式为：{type, coefficients, max_degree}。其中 type 是'p'（primal）或者'd'（dual）。系数 C 是定义劳伦多项式 P 系数的向量。max_degree 是 P 中单项的最高次数。

劳伦多项式 P 的形式为：

P(z) = C(1)*z^d + C(2)*z^(d−1) + ... + C(m)*z^(d−m+1)

提升方案 LS 中，对于 k 从 1 到 $N-1$，LS{k,:}具有 ELS 的格式：LS{k,1}是提升类型'p' (primal)或者'd'(dual)。LS{k,2}是相应的提升滤波器。LS{k,3}是和 l 滤波器 LS{k,2}相关的劳伦多项式的最高次数。LS{N,1}是 prima 的归一化（实数）。LS{N,2}是 dual 的归一化（实数），即 LS{N,1}*LS{N,2} = 1，LS{N,3}没有被使用。

4. bswfun 函数

【函数功能】双正交尺度和小波函数。

【语法格式】

① [PHIS,PSIS,PHIA,PSIA,XVAL] = bswfun(LoD,HiD,LoR,HiR)

② [PHIS,PSIS,PHIA,PSIA,XVAL] = bswfun(LoD,HiD,LoR,HiR,ITER)

③ [PHIS,PSIS,PHIA,PSIA,XVAL] = bswfun(LoD,HiD,LoR,HiR,'plot')

④ [PHIS,PSIS,PHIA,PSIA,XVAL] = bswfun(LoD,HiD,LoR,HiR,ITER,'plot')

【使用说明】该函数使用级联算法。格式①返回两对滤波器（LoD，HiD）与（LoR，HiR）的两对尺度和小波函数（PHIA，PSIA）与（PHIS，PSIS）在网格 XVAL 上的近似值。格式②进行 ITER 次反复调用，计算两对尺度和小波函数。格式③计算并画出这些函数。

5. filt2ls 函数

【函数功能】将四联滤波器变换为提升方案。

【语法格式】

LS = filt2ls(LoD,HiD,LoR,HiR)

【使用说明】该函数返回 4 个输入滤波器 LoD、HiD、LoR 和 HiR 的提升方案 LS。

6. liftfilt 函数

【函数功能】在四联滤波器上应用基本提升方案。

【语法格式】

① [LoDN,HiDN,LoRN,HiRN] = liftfilt(LoD,HiD,LoR,HiR,ELS)

② [LoDN,HiDN,LoRN,HiRN] = liftfilt(LoD,HiD,LoR,HiR,ELS,TYPE,VALUE)

【使用说明】4 个滤波器 LoD、HiD、LoR 和 HiR 得到基本提升步骤 ELS，格式①返回由 ELS 得到的 4 个滤波器 LoDN、HiDN、LoRN 和 HiRN。

ELS 的结构如下：

TYPE = ELS.type，它包含基本提升步骤的类型。TYPE 可能的取值是'p' (primal)或者'd' (dual)。VALUE = ELS.value，它包含基本提升步骤的劳伦多项式 T。如果 VALUE 是一个向量，相关的劳伦多项式 T 相当于 laurpoly(VALUE,0)。另外，ELS 可以是一种尺度步骤，这样，TYPE 为's'(scaling)，VALUE 是尺度微分。

格式②返回同样的结果。

7. liftwave 函数

【函数功能】提供常用小波的提升方案。

【语法格式】

① LS = liftwave(WNAME)

② LS = liftwave(WNAME,'Int2Int')

【使用说明】格式①返回小波 WNAME 相关的提升方案。LS 是一个结构，而不是整数，被函数 lwt、ilwt 和 lwt2 等使用。

格式②进行整数到整数的小波变换。使用'Int2Int'产生 LS，这样如果使用[cA,cD] = lwt(X,LS)或者 Y= lwt(X,LS)，其中 X 是整数型向量，那么结果 cA、cD 和 Y 都是整数型向量。如果忽略'Int2Int'，那么 lwt 将产生实数向量。

8. wave2lp 函数

【函数功能】提供小波的劳伦多项式。

【语法格式】[Hs,Gs,Ha,Ga] = wave2lp(W)

【使用说明】该函数返回小波 W 相关的 4 个劳伦多项式。H-多项式（G-多项式）是低通多项式。对于正交小波，Hs = Ha 且 Gs = Ga。

9. wavenames 函数

【函数功能】提供用于 LWT 的小波名信息。

【语法格式】W = wavenames(T)

【使用说明】该函数返回包含小波类型 T 的单元数组，T 的有效取值包括：'all'——所有的小波；'lazy'——"lazy"小波；'orth'——正交小波；'bior'——双正交小波。

W = wavenames 等同于 W = wavenames('all')。

10. lwt 函数

【函数功能】一维提升小波变换。

【语法格式】

① [CA,CD] = lwt(X,W)

② X_InPlace = lwt(X,W)

③ lwt(X,W,LEVEL)

④ X_InPlace = lwt(X,W,LEVEL,'typeDEC',typeDEC)

⑤ [CA,CD] = lwt(X,W,LEVEL,'typeDEC',typeDEC)

【使用说明】lwt 对于给定的提升小波进行一维提升小波分解。该函数使用多项算法。格式①对向量 X 进行提升小波分解，计算出低频系数向量 cA 和高频系数向量 cD。W 是提升小波名。格式②也是计算低频和高频系数。两个系数的存储方式为 cA = X_InPlace(1:2:end) 和 cD = X_InPlace(2:2:end)。格式③进行 LEVEL 层计算提升小波分解。格式④和⑤使用提升小波，当 typeDEC = 'w' 或 typeDEC = 'wp'时进行 LEVEL 层小波或小波包分解。另外，除了给定提升小波名以外，还可以使用相关的提升方案 LS: lwt(X,LS,…) 来代替 lwt(X,W,…)。

11. lwt2 函数

【函数功能】二维提升小波变换。

【语法格式】

① [CA,CH,CV,CD] = lwt2(X,W)

② X_InPlace = lwt2(X,LS)

③ lwt2(X,W,LEVEL)

④ X_InPlace = lwt2(X,W,LEVEL,'typeDEC',typeDEC)

⑤ [CA,CD] = lwt2(X,W,LEVEL,'typeDEC',typeDEC)

【使用说明】lwt2 对于给定的提升小波进行二维提升小波分解。该函数使用多项算法。

格式①对向量 X 进行提升小波分解，计算出低频系数矩阵 CA 和高频系数矩阵 CH、CV 和 CD。W 是提升小波名。格式②也是计算低频和高频系数。格式③计算尺度为 LEVEL 时的提升小波分解。格式④和⑤使用提升小波，当 typeDEC = 'w'或 typeDEC = 'wp'时，进行尺度 LEVEL 的小波或小波包分解。

除了给定提升小波名以外，还可以使用相关的提升方案 LS: lwt2(X,LS,...) 来代替 LWT2(X,W,...)。

12. lwtcoef 函数

【函数功能】提取或重构一维 LWT 小波系数。

【语法格式】

① Y = lwtcoef(TYPE,XDEC,LS,LEVEL,LEVEXT)

② Y = lwtcoef(TYPE,XDEC,W,LEVEL,LEVEXT)

【使用说明】格式①返回由 XDEC 提取的尺度为 LEVEXT 的系数或重构系数。XDEC 是由提升方案 LS 得到的尺度为 LEVEL 层的分解。TYPE 的有效取值，如表 15.2 所示。

表 15.2　TYPE 的有效取值

TYPE 值	描　述	TYPE 值	描　述
'a'	低频	'ca'	低频系数
'd'	高频	'cd'	高频系数

格式②使用提升小波 W，返回同样的输出。

13. lwtcoef2 函数

【函数功能】提取或重构二维 LWT 小波系数。

【语法格式】

① Y = lwtcoef2(TYPE,XDEC,LS,LEVEL,LEVEXT)

② Y = lwtcoef2(TYPE,XDEC,W,LEVEL,LEVEXT)

【使用说明】格式①返回由 XDEC 提取的尺度为 LEVEXT 的系数或重构系数。XDEC 是由提升方案 LS 得到的尺度为 LEVEL 的分解。TYPE 的有效取值，如表 15.3 所示。

表 15.3　　TYPE 的有效取值

TYPE 值	描　　述	TYPE 值	描　　述
'a'	低频	'ca'	低频系数
'h'	水平高频	'ch'	水平高频系数
'v'	垂直高频	'cv'	垂直高频系数
'd'	对角高频	'cd'	对角高频系数

格式②使用提升小波 W，返回同样的输出。

14. ilwt 函数

【函数功能】一维提升小波反变换。

【语法格式】

① X = ilwt(AD_In_Place,W)

② X = ilwt(CA,CD,W)

③ X = ilwt(AD_In_Place,W,LEVEL)

④ X = ilwt(CA,CD,W,LEVEL)

⑤ X = ilwt(AD_In_Place,W,LEVEL,'typeDEC',typeDEC)

⑥ X = ilwt(CA,CD,W,LEVEL,'typeDEC',typeDEC)

【使用说明】ilwt 使用指定的提升小波进行提升小波重构。

格式①使用提升小波重构得到的低频和高频系数向量 AD_In_Place，计算重构向量 X。W 是提升小波名（参见 liftwave 函数）。格式②使用提升小波重构得到的低频系数向量 CA 和高频系数向量 CD，计算重构向量 X。格式③和④计算 LEVEL 层的提升小波重构。格式⑤和⑥使用提升小波，在 typeDEC = 'w'或 typeDEC = 'wp'时，分别计算 LEVEL 层时的小波或小波包分解。

除了使用提升小波以外，还可以使用相关的提升方案 LS: X = ilwt(…,LS, …)来代替 X = ILWT(…,W,…)。

15. ilwt2 函数

【函数功能】二维提升小波反变换。

【语法格式】

① X = ilwt2(AD_In_Place,W)

② X = ilwt2(CA,CH,CV,CD,W)

③ X = ilwt2(AD_In_Place,W,LEVEL)

④ X = ilwt2(CA,CH,CV,CD,W,LEVEL)

⑤ X = ilwt2(AD_In_Place,W,LEVEL,'typeDEC',typeDEC)

⑥ X = ilwt2(CA,CH,CV,CD,W,LEVEL,'typeDEC',typeDEC)

【使用说明】ilwt2 使用指定的提升小波进行提升二维小波重构。

格式①使用提升小波重构得到的低频和高频系数矩阵 AD_In_Place，计算重构矩阵 X。W 是提升小波名（参见 liftwave 函数）。格式②使用提升小波重构得到的低频系数矩阵 CA 和高频系数矩阵 CH、CV 和 CD，计算重构矩阵 X。格式③和④计算尺度为 LEVEL 时的提升小波重构。格式⑤和⑥使用提升小波，在 typeDEC = 'w'或 typeDEC = 'wp'时，分别计算尺度 LEVEL 时的小波或小波包分解。

除了使用提升小波以外，还可以使用相关的提升方案 LS: X = ilwt2(…,LS,…)来代替 X = ILWT2(…,W,…)。

16. laurmat 函数

【函数功能】劳伦矩阵类 LM 的构造器。

【语法格式】M = laurmat(V)

【使用说明】该函数返回劳伦矩阵对象 M，V 可以是劳伦多项式的单元阵列（最多二维）或者是普通的矩阵。

17. laurpoly 函数

【函数功能】劳伦多项式类 LP 的构造器。

【语法格式】

① P = laurpoly(C,d)

② P = laurpoly(C,'dmin',d)

③ P = laurpoly(C,'dmax',d)

【使用说明】格式①返回劳伦多项式对象。C 是向量，它的元素是多项式 P 的系数。d 是 P 的最高次数。

如果 m 是向量 C 的长度，P 代表以下的劳伦多项式：

P(z) = C(1)*z^d + C(2)*z^(d−1) + …+ C(*m*)*z^(d−*m*+1)

格式②指定的是 P 的最低次数，而不是最高次数。相应的输出 P 代表以下劳伦多项式：

P(z) = C(1)*z^(d+*m*−1) + … + C(*m*−1)*z^(d+1) + C(*m*)*z^d

P = laurpoly(C,'dmax',d) 等同于 P = laurpoly(C,d)。

15.3　Matlab 提升小波变换的实现技术

15.3.1　一维提升小波变换

例程 15-1

f15_1.m

```
% 使用 Haar 小波，得到相应的提升方案
```

```
lshaar = liftwave('haar');
% 添加 ELS 到提升方案
els = {'p',[-0.125 0.125],0};
lsnew = addlift(lshaar,els);
% 对于简单信号，尺度为 1 进行 LWT
x = 1:8;
[cA,cD] = lwt(x,lsnew)
% 对上面的信号，进行整数 LWT
lshaarInt = liftwave('haar','int2int');
lsnewInt = addlift(lshaarInt,els);
[cAint,cDint] = lwt(x,lsnewInt)
```

结果如下：

```
cA =
    1.9445    4.9497    7.7782    10.6066
cD =
    0.7071    0.7071    0.7071    0.7071
cAint =
    1    3    5    7
cDint =
    1    1    1    1
```

15.3.2 二维提升小波变换

例程 15-2

f15_2.m

```
% 使用 Haar 小波，得到相应的提升方案
lshaar = liftwave('haar');
% 添加 ELS 到提升方案
els = {'p',[-0.125 0.125],0};
lsnew = addlift(lshaar,els);
% 对于简单图像，进行 1 层 LWT
x = reshape(1:16,4,4);
[cA,cH,cV,cD] = lwt2(x,lsnew)
% 对上面的图像，进行整数 LWT
lshaarInt = liftwave('haar','int2int');
lsnewInt = addlift(lshaarInt,els);
[cAint,cHint,cVint,cDint] = lwt2(x,lsnewInt)
```

结果如下：

```
cAint =
    3    11
    5    13
cHint =
    1    1
```

```
     1     1
cVint =
     4     4
     4     4
cDint =
     0     0
     0     0
```

15.3.3　一维提升小波反变换

例程 15-3

f15_3.m

```
% 使用 Haar 小波，得到相应的提升方案
lshaar = liftwave('haar');
% 添加 ELS 到提升方案
els = {'p',[-0.125 0.125],0};
lsnew = addlift(lshaar,els);
% 对于简单信号，尺度为 1 进行 LWT
x = 1:8;
[cA,cD] = lwt(x,lsnew);
% 对上面的信号，进行整数 LWT
lshaarInt = liftwave('haar','int2int');
lsnewInt = addlift(lshaarInt,els);
[cAint,cDint] = lwt(x,lsnewInt);
% 进行逆变换
xRec = ilwt(cA,cD,lsnew);
err = max(max(abs(x-xRec)))
xRecInt = ilwt(cAint,cDint,lsnewInt);
errInt = max(max(abs(x-xRecInt)))
结果如下：
err =
  4.4409e-016
errInt =
     0
```

15.3.4　二维提升小波反变换

例程 15-4

f15_4.m

```
% 使用 Haar 小波，得到相应的提升方案
lshaar = liftwave('haar');
% 添加 ELS 到提升方案
els = {'p',[-0.125 0.125],0};
```

```
lsnew = addlift(lshaar,els);
% 对于简单图像，尺度为 1 进行 LWT
x = reshape(1:16,4,4);
[cA,cH,cV,cD] = lwt2(x,lsnew);
% 对上面的图像，进行整数 LWT
lshaarInt = liftwave('haar','int2int');
lsnewInt = addlift(lshaarInt,els);
[cAint,cHint,cVint,cDint] = lwt2(x,lsnewInt);
% 进行逆变换
xRec = ilwt2(cA,cH,cV,cD,lsnew);
err = max(max(abs(x-xRec)))
xRecInt = ilwt2(cAint,cHint,cVint,cDint,lsnewInt);
errInt = max(max(abs(x-xRecInt)))
结果如下：
err =
      0
errInt =
      0
```

参 考 文 献

[1] 张贤达. 非平稳信号分析与处理. 北京: 国防工业出版社, 1999

[2] 胡晓东, 曹立明, 施鸿宝. 基于小波包分解的多尺度图像边缘检测算法及实现. 上海铁道大学学报, 1999,20（10）: 43～48

[3] 邢庆滨, 孙圣和. 基于小波包理论的 ECG 信号数据压缩. 中国医学物理学杂志, 1997,14（4）: 212-216

[4] 赵明, 朱春喜. 小波包理论及其在图像处理中的应用. 益阳师专学报, 1998, 15（5）:4～7

[5] 潘润红, 马平清等. 利用时频分析进行合成孔径雷达运动目标成像的研究. 电子科学学刊, 1998, 20（3）: 392～296

[6] 张子瑜, 吴镇扬等. 基于 Hough 变换的任意时频分布线条特征提取. 电子学报, 2001, 29（4）: 433～435

[7] 王衍文, 王海滨, 程敬之. 心音信号的时频分析. 生物医学工程学杂志, 1998,15（1）: 41～46

[8] 刘林, 郝保国. 时频分析理论和应用. 计算机自动测量与控制, 2001,9（4）: 44～46

[9] 吴小培, 冯焕清等. 基于小波变换的脑点瞬态信号检测. 数据采集与处理, 2001, 16（1）: 86～89

[10] 王衍文, 王海滨, 程敬之. 广义时频分析及其在 PCG 信号分析中的应用. 生物医学工程学杂志, 1998,17（2）: 117～125

[11] 荆双喜, 铁占绪, 张英琦. 基于小波分析的机械故障诊断技术研究. 煤炭学报, 2000,25: 143～146

[12] 任国全, 韦有民, 郑海起. 基于小波分析的轴承故障诊断研究. 河北省科学院学报, 2002, 19（2）: 112～116

[13] 刘华, 蔡正敏等. 小波包算法在滚动轴承的在线故障诊断中的应用. 机械科学与技术, 1999,18（2）: 301～303

[14] 张力, 丁常富, 韩中合. 小波分析在振动信号处理领域中的应用. 电力情报, 2001,3: 5～7

[15] 何岭松. 小波函数性质及其对小波分析结果的影响. 振动工程学报, 2000,13（1）: 143～146

[16] Time-Frequency Toolbox for use with MATLAB——Reference Guide

[17] Time-Frequency Toolbox for use with MATLAB——Tutorial

[18] http://www.mathworks.com.

[19] 秦前清, 杨宗凯编著. 实用小波分析. 西安: 西安电子科技大学出版社, 1994

[20] 胡昌华, 张军波等编著. 基于 MATLAB 的系统分析与设计——小波分析. 西安: 西安电子科技大学出版社, 1999

[21] 程正兴著. 小波分析算法与应用. 西安: 西安交通大学出版社, 1998

[22] 杨福生著. 小波变换的工程分析与应用. 北京: 科学出版社, 1999
[23] 彭玉华著. 小波变换与工程应用. 北京: 科学出版社, 1999
[24] Wavelet Toolbox User's Guide. Mathworks Inc, 2004
[25] 飞思科技产品研发中心著. Matlab6.5 辅助小波分析与应用. 北京：电子工业出版社, 2003,1
[26] 飞思科技产品研发中心著. Matlab6.5 辅助图像处理与应用. 北京：电子工业出版社, 2003,1
[27] 飞思科技产品研发中心著. 小波分析理论与 MATLAB7 实现. 北京：电子工业出版社, 2005,3
[28] 曲守平, 张立军. 路面不平度的时频小波分析. 长春大学学报, 1999, 9（1）: 5～8
[29] 杨绍奎. 用时间序列分析法描述路面不平度. 兵工学报, 1996,1:1～7
[30] 赵珩, 卢士富. 路面对四轮汽车输入的时域模型. 汽车工程, 1999,21（2）:112～117
[31] 王岩松, 耿艾莉, 张立军. 基于小波变换的四轮车辆非平稳振动时频研究. 汽车工程,2004,26（2）:177～202